Analysis of Positron Lifetime Spectra in Polymers

Jag J. Singh,
Gerald H. Mall,
and Danny R. Sprinkle
Analysis of Positron Lifetime Spectra in Polymers

Jag J. Singh
Langley Research Center
Hampton, Virginia

Gerald H. Mall
Computer Sciences Corporation
Hampton, Virginia

Danny R. Sprinkle
Langley Research Center
Hampton, Virginia
Summary

A new procedure for analyzing multicomponent positron lifetime spectra in polymers has been developed. It requires initial estimates of the lifetimes and intensities of various components, which are readily obtainable by a standard spectrum stripping process. These initial estimates, after convolution with the timing-system-resolution function, are then used as the inputs for a nonlinear least-squares analysis to compute the estimates that conform to a global-error minimization criterion. The convolution integral uses the full experimental resolution function, in contrast to the previous studies in which analytical approximations of it were utilized. These concepts have been incorporated into a generalized computer program for analyzing positron lifetime spectra (PAPLS) in polymers. The validity of this program has been tested by using several artificially generated data sets. These data sets were also analyzed with the widely used POSITRONFIT program. In almost all cases, the PAPLS program gives closer fits to the input values. The new procedure has been applied to the analysis of several lifetime spectra measured in metal-ion containing Epon 828 samples. The results are described in this report.

Introduction

Positrons are slowed down quickly (refs. 1 to 4) upon entering a condensed medium such as polymers. After thermalization, they may annihilate as free positrons or after being trapped in microvoids or at defect sites on molecular chains. Some of the trapped positrons also loosely bind with the molecular electrons to form positronium (Ps) atoms. (See refs. 3 to 8.) Depending on the final states of the positrons they annihilate at different times. The lifetime of the positron can thus provide valuable information about the electronic and physical structure of the bulk material and the various defects or impurities in it.

Positron lifetime studies make use of radioactive sources that emit a gamma ray almost simultaneously with the positron. The widely used Na$^{22}$ source emits a 1.28-MeV photon within 3 psec of the emission of the preceding positron. The detection of this gamma photon marks the zero time (reference time) for a lifetime measurement, since the thermalization time for the positrons in polymers is much less than their subsequent lifetime and can usually be ignored. (See refs. 9 to 11.) The annihilation of the positron is signalled by the detection of one of the two annihilation photons produced in the $e^+ + e^- \rightarrow 2h\nu$ (511 keV each) process. The time interval between these two signals is a measure of the positron lifetime and is commonly measured by fast timing systems. The two widely used timing systems are described in references 12 to 15.

A typical lifetime spectrum of positrons annihilating in polymers (refs. 7 and 16 to 19) is expected to consist of at least three exponentially decaying components, each characterized by a mean lifetime and a relative probability amplitude. The three components are short-lifetime (resulting from prompt positron and parapositronium decays), intermediate-lifetime (resulting from trapped positron decays), and long-lifetime (resulting from orthopositronium decays). Quite often, the lifetime spectra in polymers have been analyzed by considering only two components, representing the mean lives of the “shorter” and “longer” components, for reasons of simplicity and/or expediency. (See refs. 7 and 20 to 28.) However, a detailed analysis that provides at least three components is preferable, since the annihilations of all the positron groups are dependent on their local atomic environments and can thus provide more detailed information about the host-material defect structure.

Observed spectra are subject to a finite time resolution of the lifetime measurement system. Data events which would theoretically fall into a certain time interval (channel) actually exhibit a near Gaussian distribution over many channels depending on the timing-system resolution. In previous studies, the timing-system-resolution function has been approximated by an analytical expression for reasons of simplicity and convenience of analysis. However, the resolution function does not lend itself to a complete description by a reasonable analytical expression. It would therefore be preferable to use the full experimental resolution function for deconvolving the lifetime spectra.

The purpose of this report is to develop a procedure for resolving a multicomponent experimental spectrum into individual lifetime components while imposing as few constraints and assumptions as possible. This procedure has been incorporated into a computer program for analyzing positron lifetime spectra (PAPLS) in polymers. This program has been used to analyze lifetime spectra observed in metal-ion containing Shell Epon 828 epoxy samples, and the results are compared with those obtained with the previously developed analysis techniques. This procedure is adaptable for other types of condensed matter within which positronium is likely to be formed.
Problem Statement

The lifetime spectra of positrons annihilating in a molecular solid consist of several components. It is often convenient to group the several lifetimes into three broad categories. (See ref. 29.) The shortest lifetime component (0.1 to 0.5 nsec) includes prompt free-positron decay and parapositronium decay. The intermediate lifetime component (0.5 to 0.9 nsec) is believed to arise from the annihilation of trapped positrons. The longest lifetime component (1 to 4 nsec) arises from pick-off annihilation of orthopositronium atoms. Each of the annihilation processes contributes a decaying exponential, with a characteristic decay constant, to the spectrum. In the previous studies of positron annihilation spectroscopy (PAS) in molecular solids (refs. 2, 7, and 21 to 28), it has often been convenient to resolve the experimental lifetime spectra into two components. This procedure has been justified on the basis of the arguments that the long-life component, which results from the pick-off annihilation of the triplet positronium and is the most sensitive indicator of the host-material properties, can be easily deconvolved from the rest of the spectrum; the short and intermediate lifetime components are then lumped together into a compound short-life component. It would, however, be more appropriate if all the positron groups could be deconvolved (refs. 16 to 19) particularly since the other groups are also dependent on the atomic environment of the trapped positrons and can provide some very useful information about the host defect structure.

For a three-component system, the lifetime spectrum can be written as the sum of three exponentials as follows:

\[ n(t) = \sum_{i=1}^{3} A_i e^{-\lambda_i t} + B \]

where \( A_i \) and \( \lambda_i \) represent zero-time amplitudes and decay constants, respectively, and \( B \) is a constant background.

When experimental \( n(t) \) versus \( t \) data are displayed on a semilog format, the spectrum appears to be a sum of three straight lines—each representing an exponential decay. The three groups are easily discernible if \( \lambda_1, \lambda_2, \) and \( \lambda_3 \) are sufficiently different from each other and if their relative probabilities are comparable. In that case, the spectrum can be decomposed into individual components rather easily. Sometimes, though, the \( \lambda_i \) values are not sufficiently different, and the longer lived components are weaker; then, the spectrum analysis is a difficult problem.

The theoretical spectrum defined by equation (1) must be modified for the finite time resolution of the lifetime measurement system. Data events which would theoretically fall in a certain time interval (channel) actually exhibit a near Gaussian distribution whose effective full width at half maximum (FWHM) equals the timing-system resolution. The form of the time-resolution function can be determined by measuring a prompt spectrum, such as that obtained from detecting the near-simultaneous 1.17-MeV and 1.33-MeV gamma photons from a Co 60 source, with the same experimental settings for electronics that are used in the actual lifetime experiment. Zero time is determined by the centroid of this prompt spectrum. Incorporation of the finite-resolution function into equation (1) modifies the lifetime spectrum as follows:

\[ n(t) = \int_{-\infty}^{+\infty} R(t - t') n(t') \, dt' \]  

where \( R(t - t') \) represents the timing-system resolution-function and \( t' \) is the time variable, which differs from \( t \) by an integral number of time channels.

In previous studies, the resolution function has been assumed to be Gaussian; therefore, a direct analytical deconvolution of the observed data was possible. This approximation, however, tended to introduce systematic errors and to give poorer fits to the lifetime values. To improve on this situation, several authors have used modified Gaussian representations of the resolution function. Lichtenberger et al. (refs. 30) used a double-sided exponential function, whereas Hall et al. (ref. 31) used a side-sloped function to represent the resolution function. Eldrup et al. (refs. 7 and 8) and Kirkegaard et al. (ref. 32), on the other hand, assume the time-resolution function to be a sum of three Gaussian functions rather than a single Gaussian. Use of these various forms of analytical approximations improves the quality of the fits without complicating the solution of the convolution integral.

However, it has long been recognized that the experimental resolution functions of the fast timing systems used in PAS studies have slowly decaying characteristics which do not admit to simple analytical representations. To overcome the limitations imposed by imperfect analytical representation of the resolution function, actual experimental resolution functions are used as the smearing functions. An additional argument in support of this procedure arises from the fact that each timing system has its own unique resolution function. Even though this
The approach makes the analysis procedure slightly more tedious, it is better suited for analyzing experimental spectra with inadequate statistics, as is often the case with the more promising types of polymers such as PMR-15, LARC-TPI, and other high-temperature polyimides. The use of the actual experimental resolution function, rather than an approximate analytical representation of it, constitutes the major point of difference between the present and previous investigations. However, because of the numerical representation of the resolution function, it is not possible to analytically convolve the instrumental resolution effects into the theoretical spectrum. They have to be incorporated numerically as illustrated in the following example.

Assume that a lifetime spectrum has the following three components:

\[ \tau_1 = 457 \text{ psec}; \quad I_1 = 79.7\% \]
\[ \tau_2 = 917 \text{ psec}; \quad I_2 = 5.3\% \]
\[ \tau_3 = 2300 \text{ psec}; \quad I_3 = 15.0\% \]

where \( \tau_i \) is the lifetime of the \( i \)th component and \( I_i \) is the intensity of the \( i \)th component. Also assume that the timing-system resolution is 250 psec, the time channel width is 61.23 psec, and the total counting time is such that the peak channel count is \( 10^5 \) and the background count is 10. Figure 1 shows the histogram that is observed with a perfect system. Random statistical fluctuations have been included in this calculated histogram. A timing system with a finite resolution will smear this histogram into a spectrum shown in figure 2. The total counts which fall in any one channel for a perfect system are spread over an experimental distribution defined by the Co\(^{60} \) spectrum. This smearing process is repeated for each time channel in the theoretical histogram. Finally, the sums of all redistribution counts in the individual channels are recorded to give the computed spectrum (fig. 2) that would be observed with a finite-resolution timing system.

Close comparison of figures 1 and 2 shows that the first 10 or so channels are severely disturbed by the finite-resolution effects. Channels higher than 10 are essentially undisturbed, which indicates that the counts gained from the neighboring channels neutralize the losses to the neighboring channels that resulted from the finite-resolution effects. Figure 2 also shows that the peak in the synthesized smeared spectrum is about 100 psec to the right of the time zero \( (t_z) \) (reference) channel. This information can also be used to help locate the zero of the time scale in subsequent analyses. Figure 3 shows an actual experimental lifetime spectrum of positrons in an epoxy target. Obviously, the spectra shown in figures 2 and 3 are similar.

In the following sections, a procedure for resolving the experimental lifetime spectrum into three or more components is described. Previous attempts to develop procedures and programs for analyzing multicomponent positron lifetime spectra are described in references 32 to 34.

**Spectrum Analysis**

**Procedure**

The analysis procedure is described by referring to the experimental data shown in figure 3(b). The data have been plotted on a semilog paper in order to simplify the discussion. A cursory examination of this figure shows that it has three distinct components superimposed on a constant background. The stripping process starts by subtracting the background, which can be easily determined by averaging about 50 channels to the far right where no genuine positron decay events contribute to the observed data. Figure 4(a) shows the spectrum remaining after the background is subtracted. (The time-scale origin \( t_z \) has been shifted to channel zero and the abscissa has been expanded for the sake of clarity and for convenience of discussion.) Attention is first focused on the longest lifetime component (i.e., the third component). The first and second components do not seem to make any contributions to the data beyond channel 150. It would therefore appear that data in channels 150 to 180 are solely attributable to the third component, the initial estimate of which can be obtained by a least-squares fit to the data in these channels. After subtracting the third component from the background-free spectrum, the remaining spectrum (fig. 4(b)) has two distinct components. The counts in channels 40 to 60 are reasonably free from the effects of the first component. A least-squares fit to the data in channels 40 to 60 then provides the necessary initial information about the second component. Figure 4(c) illustrates the fit for the second component. The spectrum remaining after subtracting the second component is then solely attributable to the first component. Figure 4(d) shows this residual spectrum. Analysis of the residual spectrum then provides an initial estimate for the first component in the spectrum. Figure 4(e) illustrates the fit for the first component. After obtaining initial estimates for the three components, the finite-resolution effects are included in each component as discussed in the preceding section. The sum of these modified components gives the initial computed least-squares spectrum for comparison with the experimental spectrum.
Figure 4(f) illustrates the initial computed spectrum. This initial computed spectrum is then reiterated until the difference between the computed and the experimental spectra is minimized. The final comparison between the fitted and experimental spectra is shown in figure 4(g). A computer program developed by using these procedures is described in appendixes A and B.

Test of New Method

To validate the computational procedure described in appendixes A and B, several artificial sets of data with random fluctuations were first constructed. An experimentally observed $\text{Co}^{60}$ spectrum was used to convolve the input sum of exponentials for simulation of the true experimental data. These simulated spectra were then analyzed using the same procedure as for the actual data. A second experimental $\text{Co}^{60}$ spectrum was used as the input for the resolution function for PAPLS and POSITRONFIT analyses. Typical results for two such cases are summarized in tables I to III.

It is apparent from the results summarized in tables II and III that the present method deconvolves the artificial spectra correctly. It is also apparent that PAPLS provides closer fits to the input values for almost all the parameters than the POSITRONFIT program, particularly in the case of intensity values for the three components. Figures 5(a) and 5(b) illustrate the comparison between the fits predicted by PAPLS and POSITRONFIT programs, respectively, for case 1.

Applications

The PAPLS program has been applied in the analysis of positron lifetime spectra observed in metal-ion containing Epon 828 epoxy samples. The target specifications and experimental lifetime spectra observed in them are listed in appendix C. The same spectra were also analyzed using the POSITRONFIT program for the purpose of comparison. The results of this analysis are also included in table IV. A comparison of the two sets of values shows that while they are in general agreement, certain differences do exist. For example, most of the POSITRONFIT lifetime values are slightly higher than the values given by the PAPLS program.$^1$ Similarly, the intensities of all the POSITRONFIT shortest lifetime components are higher, whereas the intensities of the intermediate lifetime components are lower than the corresponding PAPLS values. These differences are attributable to the fact that the PAPLS program uses the exact resolution function, whereas the POSITRONFIT program assumes it to be a Gaussian or sum of Gaussians. The resolution function is not rigorously representable by a sum of Gaussians or Gaussians and exponentials.

Because of the manner in which $t_z$ is calculated in PAPLS, it takes considerably longer to complete a PAPLS analysis than a POSITRONFIT analysis. Overall, however, the PAPLS program is more accurate, because it uses an exact resolution function rather than an approximate analytical representation.

Concluding Remarks

A new technique for analyzing multicomponent positron lifetime spectra in polymers has been developed. It utilizes the actual experimental timing-system-resolution function, rather than an analytical approximation, for deconvolving the experimental spectra. These concepts have been incorporated into a computer program for analyzing positron lifetime spectra (PAPLS) in polymers and other condensed media where positronium is likely to be formed. The validity of this program has been tested using several artificially generated data sets with random statistical fluctuations. Typically, the fitted parameters agree with the input values within ±2 percent. These same data sets were also analyzed using the widely used POSITRONFIT program for the purpose of comparison. In almost all cases, the present technique gives closer fits to the input values. Both PAPLS and POSITRONFIT have been used to analyze several lifetime spectra measured in metal-ion containing Epon 828 epoxy samples. Even though the results obtained by the two programs are essentially equal, the PAPLS results are considered more accurate, because PAPLS does not make an a priori assumption about the nature of the life timing-system-resolution function.

1 The PAPLS program gives slightly higher $\tau_2$ and $\tau_3$ values in Epon 828 containing 0.1 mole fraction of $\text{Cr} (\text{DMSO})_6 (\text{ClO}_4)_3$.

NASA Langley Research Center
Hampton, Virginia 23665-5225
October 11, 1988
Appendix A

Computer Program (PAPLS)

The computer program for analyzing positron lifetime spectra (PAPLS) is written in FORTRAN Version 5 language for the Control Data CYBER 170 series digital computer system with the Network Operating System (NOS 2.4). Machine dependence is limited to the use of several library routines (matrix inversion and graphics routines), and the program should be readily adaptable to other computer systems. All routines not included in the program are described in the code. The program requires approximately 70,000 octal locations of core storage, and a typical analysis requires approximately 100 central processing unit (CPU) seconds.

It is assumed in the program that the positron lifetime spectrum can be described by the equation

\[
n_k = \sum_{j=0}^{\infty} W_{j,k} \sum_{i=1}^{3} A_i e^{-\lambda_i (t_j - t_z)} + B \quad (A1)
\]

where \( n_k \) is the number of counts in channel \( k \), \( A_i \) is the amplitude at time zero \( t_z \) of each component, \( \lambda_i = 1/\tau_i \), \( \tau_i \) is the lifetime of each component, \( t_j \) is the time corresponding to channel \( j \), \( B \) is a constant background, and the values of \( W_{j,k} \) are normalized weights representing the resolution function smear- ing between channels \( j \) and \( k \). In this expression, the values of the amplitudes \( A_i \) are zero for \( t < t_z \). The number of components is arbitrary, but the particular version of the program being described has three components. The weights \( W_{j,k} \) may be defined by using either an experimental resolution function or an analytical representation. The program defaults to a Gaussian time function if an experimental spectrum is not provided. The default weights are computed by

\[
W_{j,k} = \frac{1}{\sqrt{\pi} \sigma} e^{-(t_j - t_z)^2/\sigma^2} \quad (A2)
\]

where \( \sigma \) is the standard deviation.

The process of obtaining a solution begins by first eliminating the background from the spectrum. The background is obtained by averaging counts over several channels (the default is 51 channels) at the high end of the spectrum. Thus, it is important that enough channels are provided so that the contribution of genuine positron decay events is negligible within the region used for background averaging.

Before applying the least-squares technique, described in detail in appendix B, to equation (A1) with \( B \) subtracted from both sides of the expression, it is necessary to estimate initial values for \( A_i \) and \( \lambda_i \). Rather crude estimates are usually sufficient when the experimental data are relatively noise free. A stripping process is used to obtain these estimates. For this purpose, equation (A1) is approximated by the following simplistic expression:

\[
\ln(n - B) = \ln(A_1) - \lambda_1 t + \ln(A_2) - \lambda_2 t + \ln(A_3) - \lambda_3 t \quad (A3)
\]

At this point of the solution process, it is further assumed that the resolution is perfect and that the \( z \) is exactly at the center of a channel a fixed number of channels (the default is 2 channels) to the left of the observed spectrum maximum.

The longest lifetime component is first addressed by examining only the data in a range where the contribution from the other two components has decayed to a negligible value (the default range is from channel 100 to channel 130 to the right of \( t_z \)). The data in these channels are least-squares fitted to a straight line to provide the initial estimates for \( A_3 \) and \( \lambda_3 \) using these values, the third component is subtracted from equation (A3). The intermediate lifetime component is next considered by examining only the data in a range where the short lifetime component has decayed to a small value (the default range is from channel 38 to channel 58 to the right of \( t_z \)). As before, a straight line fit yields the initial estimates for \( A_2 \) and \( \lambda_2 \). The intermediate lifetime component is next subtracted from equation (A3). Finally, initial estimates for \( A_1 \) and \( \lambda_1 \) are obtained by a straight line fit to the remaining data. The default range for this fit is from channel 11 through channel 20 to the right of \( t_z \).

With these six values used as initial estimates, an iterative process is initiated to determine the solution of equation (A1). The first step is to hold these six values constant and find the value of \( t_z \), which minimizes the sum of the squares of the residuals. The entire range of data is used starting eight channels to the right of the peak. The initial estimates of \( A_i \) and \( \lambda_i \) are used with this value of \( t_z \), and the least-squares technique is employed to generate new values for \( A_i \) and \( \lambda_i \). For this portion of the solution, the range of data used does not include data from eight channels to the left of the peak to three channels to the right of the peak. The improved values of \( A_i \) and \( \lambda_i \) are then held fixed, and \( t_z \) is once more adjusted to minimize the sum of the squares of the residuals. This iterative process continues until \( \Delta t_z < 10^{-3} \) channel, where \( \Delta t_z \) is the change in \( t_z \) from one iteration to the next.

The resulting seven values of \( A_i \), \( \lambda_i \), and \( t_z \) are then used as initial estimates for one final iterative
procedure. This procedure is identical to the previous step, except that all the data are used when employing the least-squares technique.

Program Input

All program control input is provided through the NAMELIST name INDAT. A summary of the input variables is provided below. (Default values are provided in parentheses.)

NDIV (51):
Number of channels at high end of spectrum to be used in obtaining average background.

BKG (computed on basis of NDIV):
Average background. If specified, it will override the computed value.

TSF (61.27):
Time-scale factor in psec per channel.

NPO (120,38,11):
Array that provides channel offsets for defining regions used in obtaining initial estimates.

NPL (30,20,9):
Array that provides interval lengths for defining regions used in obtaining initial estimates. Defaults are based on the default value of TSF.

IDCHAN (-2):
Correction factor to compensate for apparent shift in peak location that results from time-resolution function. In general, the true value of \( t_z \) is to the left of the peak in the experimental data. An examination of the program output should verify that the computed and experimental data peak in the same channel; if not, an appropriate value of IDCHAN will correct for this.

IGAUSS (1):
Flag to indicate if a Gaussian time-resolution function is to be used in fit. If IGAUSS = 0, no time resolution function is used.

SIGINV (0.135621):
Value of \( \frac{1}{\sigma^2} \), where \( \sigma \) is standard deviation of Gaussian time-resolution function.

ITERMX (30):
Maximum number of iterations to be allowed in least-squares fit.

IPRFLG (0,0,1,0,0,1):
Array controlling quantity of printed output. Each of six flags controls a specific output item. If the flag is "1," the item is printed. If the flag is "0," the item is not printed. The flags correspond to the following output items:
1. Raw data
2. Background corrected data
3. Initial-estimate fit results
4. Spectrum with initial fit subtracted
5. Fit using initial estimates
6. Final fit

IPLFLG (1,0,1,1,1,1):
Array that controls the quantity of graphic output. Each of six flags controls a specific plot. If the flag is "1," the plot is generated. If the flag is "0," the plot is not generated. The flags correspond to the same items as those for IPRFLG.

IPEEK (0,0):
Array used to simply examine data. If IPEEK(1) ≠ 0, the program produces a plot of the log of the data versus channel number; the plot begins with channel number IPEEK(1) to the right of the peak and continues through channel number IPEEK(2) to the right of the peak. No fit is attempted. This feature is useful for determining the ranges used for obtaining initial estimates.

In addition to program control input, the program reads the spectrum from TAPE1. The first record on this file should contain the number of channels and a data identification label in I4, A7 format. No more than 800 channels are allowed. The remaining records on this file contain integer counts for the number of channels specified in list-directed format. When an experimental resolution function spectrum is provided, these data complete the contents of TAPE1, which is also in list-directed format.

Program Output

The bulk of the program output is controlled by user-defined flags as indicated in the section "Program Input." The initial output (fig. A1) is always produced and provides an echo of the input spectrum with the data identification label, the number of channels, the value of the average background, and the number of channels used to compute the average background. If desired, the raw data are then output, as illustrated in figure A2, and give the time in psec, the counts, and the log of the counts for each channel. The data that are corrected for background may also be printed in the same four-column format. Data relating to the spectrum peak are then output, as shown in figure A3, and give the maximum value of the background-corrected spectrum, the channel where the peak occurs, and the channel corresponding to \( t_z \). For each component, a summary of the initial-estimate calculations can be produced as shown in figure A4. This summary identifies the component, the channel range used to obtain the estimates, and the values (with errors in parentheses) for the slope \( \lambda \), lifetime TAU, intercept ln(A),
and the area under the resulting curve. Also output are the fit standard deviation, the number of iterations required for convergence, and an error code. If this error code is not zero, the solution did not converge. For each component, the spectrum that results from subtracting the initial-estimate curve can also be output in a form similar to that in figure A2. After initial estimates have been determined for all six unknowns, the spectrum calculated from these values can be output for comparison with the experimental data. A sample of this output is included as figure A5; the sample provides the channel number, time (in psec), experimental data, and the fitted spectrum by using initial estimates. These data can be examined to verify the experimental data and the fit peak in the same channel. If not, the input variable IDCHAN requires adjustment. Sample output corresponding to the final-fit values is illustrated in figure A6. For each component, values are given (with errors in parentheses) for the slope λ, lifetime TAU, intensity \( \left( \frac{\text{Area}_i}{\sum_{j=1}^{n} \text{Area}_j} \right) \), intercept \( \ln(A) \), and the area under the resulting curve. Values are also provided for the fit standard deviation, the number of iterations required for convergence, and an error code. The final output is similar to figure A5, where the calculated spectrum is obtained using the final-fit values.

In addition to the printed output, the program produces graphic output, all of which is controlled by user-defined flags. The plots corresponding to the six flags are shown in figure A7. Results similar to figures A7(c) and A7(d) are produced for each component.
Program Listing

PROGRAM PAPLS(OUTPUT, INPUT, TAPE5=INPUT, TAPE6=OUTPUT, 
1 TAPE1)

C***PAPLS IS A COMPUTER PROGRAM FOR ANALYZING POSITRON 
LIFETIME SPECTRA. THE CURRENT VERSION OF THE PROGRAM 
IS LIMITED TO THREE COMPONENTS, BUT MINOR CODING 
CHANGES WOULD BE REQUIRED TO ENABLE THE PROGRAM 
TO PROCESS MORE OR FEWER COMPONENTS. AN ADVANTAGE 
OF PAPLS OVER OTHER PROGRAMS OF A SIMILAR NATURE 
is its handling of the time resolution function. 
The resolution function is described in tabular 
form, thus allowing any analytic (or non-analytic) 
representation. For a non-analytic function, a low 
level of noise in the representation of the time 
resolution is required for reliable program 
execution.

C***COMMON/INDAT/ICNT(800),NPTS,NPTSS,BKG,NDIV,IMAXC,TSF,IDCHAN, 
1 IPRFLOG(10), IPLFLOG(10), NPO(4),NPL(4),LABEL,CFRAC 
COMMON/PEEK/IPEEK(2) 
COMMON/GAUSDAT/AMP,SIGINV,IGAUSS,GAUSS(91) 
COMMON/OTDAT/SLOPEX(4),XOFFX(4),ESLOPE(4),EXOFF(4), 
1 NPST(4),NPLST(4),TAU(4),ETAU(4),AREA(4),EAREA(4) 
COMMON/FITDAT/XX(800),YY(800) 
COMMON/PKDAT/DATA(800),DATA(800),DATA(800),DATA(800),DATA(800),DATA(800),DATA(800), 
COMMON/CLCDAT/X(800),Y(800),Y(800),Y(800), 
COMMON/CHARS/MSGPR(4) 
DIMENSION SOL(7),E(7) 
COMMON/NEWCOM/SOL,E,ITER2,1ERR,STD 
DIMENSION Z(800) 
DATA MSGPR/6H FIRST,6H SECOND,6H THIRD,6H FOURTH/ 
DATA SLOPEX,XOFFX/8*0./ 
DATA NPASS/0/ 
1 FORMAT(1H1,5X,"...INPUT SPECTRUM (",A7,")// 
2 FORMAT(51I0) 
3 FORMAT(1H0,5X,"NUMBER OF CHANNELS IN DATA FILE = ",15 
4 FORMAT(1H0,5X,"BACKGROUND AVG. = ",1PE13.5,5X,"NDIV = ",15) 
5 FORMAT(1H0,5X,"NUMBER OF COMPONENTS TO FIT = ",15) 
6 FORMAT(1H1,5X,"RAW SPECTRUM"// 
 1 5X,"CHANNEL",10X,"TIME(PS)",10X,"COUNTS",10X,"LOG(COUNTS)"")// 
7 FORMAT(1X,F10.0,7X,1PE13.5,5X,1PE13.5,5X,1PE13.5) 
8 FORMAT(1H1,5X,"BACKGROUND CORRECTED SPECTRUM"// 
 1 5X,"CHANNEL",10X,"TIME(PS)",10X,"COUNTS",10X,"LOG(COUNTS)"")// 
9 FORMAT(1H1,5X,"SPECTRUM MAXIMUM = ",1PE13.5/ 
 1 H0,5X,"CHANNEL NUMBER CORRESPONDING TO SPECTRUM MAXIMUM = ", 
 2 H0,5X,"CORRECTED CHANNEL = ",15) 
10 FORMAT(1H0,5X,"SPECTRUM MAXIMUM = ",1PE13.5/ 
 1 H0,5X,"CHANNEL NUMBER CORRESPONDING TO SPECTRUM MAXIMUM = ", 
 2 H0,5X,"CORRECTED CHANNEL = ",15) 
11 FORMAT(1H1,5X,"SPECTRUM CORRECTED FOR ",A6," FIT"// 
 1 5X,"CHANNEL",10X,"TIME(PS)",10X,"COUNTS",10X,"LOG(COUNTS)"")//
SUBROUTINE PSEUDO

READ INPUT DATA

ICT = 0
CALL DATRD
WRITE(6,1) LABEL
WRITE(6,2) (ICNT(I),I=1,NPTS)
ISUMCT = 0.
DO 20 I=1,NPTS
   ISUMCT = ISUMCT + FLOAT(ICNT(I))
20 CONTINUE
WRITE(6,17) ISUMCT
WRITE(6,3) NPTS

CALL PSEUDO
WRITE(6,4) BKG,NDIV
NFIT = 0
DO 30 I=1,4
IF(NPL(I).NE.0) NFIT = NFIT + 1
30 CONTINUE
WRITE(6,5) NFIT
C+++++
C
C    STORE SPECTRUM IN DATAY
C    STORE LOG(SPECTRUM) IN DATAYL
C    STORE CHANNEL NUMBER IN DATAX
C    STORE TIME(PS) IN DATAXT
C
C+++++
CNTTOT = 0.
DO 40 I=1,NPTS
CNTTOT = CNTTOT + ICNT(I)
DATAY(I) = FLOAT(ICNT(I))
DATAYL(I) = 0.
IF(DATAY(I).GE.1.) DATAYL(I) = ALOG(DATAY(I))
DATAX(I) = FLOAT(I)
DATAXT(I) = DATAX(I)*TSF
40 CONTINUE
C+++++
C
C    WRITE OUT RAW DATA IF REQUESTED
C
C+++++
IF(IPRFLG(1).EQ.0) GO TO 60
WRITE(6,6)
DO 50 I=1,NPTS
WRITE(6,7) DATAX(1),DATAXT(1),DATAY(1),DATAYL(1)
50 CONTINUE
60 CONTINUE
C+++++
C
C    PLOT RAW SPECTRUM IF REQUESTED
C
C+++++
IF(IPLFLG(1).EQ.0) GO TO 70
CALL DATPLT(DATAX,DATAYL,1,SLOPE,XOFF,NPL(1),
1 NPO(1),MFIT)
70 CONTINUE
IF(IPEEK(1).NE.0) GO TO 400
C+++++
C
C    ELIMINATE BACKGROUND
C
C+++++
DO 80 I=1,NPTS
DATAY(I) = DATAY(I) - BKG
DATAYL(I) = 0.
IF(DATAY(I).GE.1.) DATAYL(I) = ALOG(DATAY(I))
80 CONTINUE
C+++++
C
10
WRITE OUT BACKGROUND CORRECTED DATA IF REQUESTED

IF(IPRFLG(2).EQ.0) GO TO 100
WRITE(6,8)
DO 90 I=1,NPTS
WRITE(6,7) DATAX(I),DATABX(I),DATAY(I),DATAYL(I)
90 CONTINUE
100 CONTINUE

PLOT BACKGROUND CORRECTED DATA IF REQUESTED

IF(IPLFLG(2).EQ.0) GO TO 110
CALL DATPLT(DATAX,DATAYL,2,SLOPE,XOFF,NPL(1),
I NPO(1),MFIT)
110 CONTINUE

LOCATE CHANNEL CONTAINING SPECTRUM MAXIMUM

YMAX = 0.
DO 120 I=1,NPTS
IF(DATAY(I).LT.YMAX) GO TO 120
YMAX = DATAY(I)
IMAX = I
120 CONTINUE

SET CORRECTED CHANNEL AND SHIFT DATA
STORE DATA IN CALCULATION ARRAYS

IMAXC = IMAX + IDCHAN
NPTSS = NPTS - IMAXC + 1
DO 130 I=1,NPTSS
X(I) = FLOAT(I-1)
Y(I) = DATAYL(I+IMAXC-1)
YL(I) = DATAYL(I+IMAXC-1)
130 CONTINUE
IF(IPRFLG(1)+IPRFLG(2).EQ.0) WRITE(6,10) YMAX,IMAX,IMAXC
IF(IPRFLG(1)+IPRFLG(2).NE.0) WRITE(6,9) YMAX,IMAX,IMAXC

CALCULATE GAUSS WEIGHTS

ICOR = IMAXC - 46
GTOT = 0.
IF(IGAUSS.EQ.2) GO TO 150
DO 140 I=38,54
XI = FLOAT(I+ICOR)
GAUSS(I) = AMP*EXP(-SIGINV*(XI - FLOAT(IMAXC))**2)
140 CONTINUE
GTOT = GTOT + GAUSS(I)
140 CONTINUE
   GO TO 170
150 CONTINUE
   DO 160 I=38,54
      GTOT = GTOT + GAUSS(I)
   160 CONTINUE
   GO TO 190
170 CONTINUE
   IF(IGAUSS.EQ.1) GO TO 190
   DO 180 I=1,91
      GAUSS(I) = 0.
   180 CONTINUE
   GAUSS(46) = 1.
190 CONTINUE
   WRITE(6,18) GTOT

C+++++
C
C LOOP THROUGH DESIRED FITS
C
C+++++

MFIT = 0
   DO 240 K=1,4
      IF(NPL(K).EQ.0) GO TO 240
      MFIT = MFIT + 1
      CALL LSQ2(NPL(K),X(NPO(K)),YL(NPO(K)),SLOPE,XOFF,
2      ITER,IERR,STD,E1,E2)
      SLOPEX(MFIT) = SLOPE
      XOFFX(MFIT) = XOFF
      ESLOPE(MFIT) = E1
      EXOFF(MFIT) = E2
      TAU(MFIT) = -1./SLOPE
      ETAU(MFIT) = ETAU(MFIT)*TSF
      AREA(MFIT) = TAU(MFIT)*EXP(XOFF)
      EAREA(MFIT) = ETAU(MFIT)*TSF
   240 CONTINUE
   NPIST(MFIT) = NPO(K)
   NPLST(MFIT) = NPO(K) + NPL(K)

C+++++
C
C WRITE OUT FIT RESULTS IF REQUESTED
C
C+++++

IF(IPRFLG(3).EQ.0) GO TO 200
   WRITE(6,12) MSGPR(MFIT),NPIST(MFIT),NPLST(MFIT),SLOPE,E1,
2   TAU(MFIT),ETAU(MFIT),XOFF,E2,AREA(MFIT),EAREA(MFIT),STD,
2   ITER,IERR
200 CONTINUE

C+++++
C
C PLOT THE FIT IF REQUESTED
C
C+++++

12
IF(IPLFLG(3).NE.0) CALL DATPLT(X,YL,5,SLOPE,
1 XOFF,NPL(K),NPO(K),MFIT)
C+++++
C
C        SUBTRACT OUT THIS RESULT
C
C+++++

FN = EXP(XOFF)
DO 210 I=1,NPTSS
   Y(I) = Y(I) - FN*EXP(SLOPE*X(I))
   IF(Y(I).LT.0.) Y(I) = 0.
   YL(I) = 0.
   IF(Y(I).GT.1.) YL(I) = ALOG(Y(I))
210 CONTINUE
C+++++
C
C        WRITE OUT CORRECTED SPECTRUM IF REQUESTED
C
C+++++

IF(IPRFLG(4).EQ.0) GO TO 230
WRITE(6,11) MSGPR(K)
DO 220 I=I,NPTSS
   IF(I.LT.IMAXC)
      WRITE(6,7) DATAX(1),DATAXT(1),DATAY(I),DATAYL(1)
      J = I - IMAXC + 1
   ELSE IF(I.GE.IMAXC)
      WRITE(6,7) DATAX(1),DATAXT(1),Y(J),YL(J)
220 CONTINUE
C+++++
C
C        PLOT THE CORRECTED SPECTRUM IF REQUESTED
C
C+++++

IF(IPLFLG(4).EQ.0) GO TO 230
CALL DATPLT(X,YL,3,SLOPE,XOFF,NPL(K),
1 NPO(K),MFIT)
230 CONTINUE
240 CONTINUE
250 CONTINUE
NPASS = NPASS + 1
C+++++

C

C        RESTORE DATA
C
C+++++

DO 260 I=1,NPTS
   DATAY(I) = DATAY(I) + BKG
   DATAYL(I) = 0.
   IF(DATAY(I).GE.1.) DATAYL(I) = ALOG(DATAY(I))
260 CONTINUE
C+++++
C
C        COMPUTE FITTED SPECTRUM
C
C+++++
DO 290 I=1,NPTS
   ZZZ = 0.
   YY(I) = 0.
   XX(I) = FLOAT(I)
   IF(I.LT.IMAXC) GO TO 280
   ARGX = DATAX(I) - (FLOAT(IMAXC) + CFRAC)
   IF(I.EQ.IMAXC) ARGX = 0.5*(0.5 - CFRAC)
   IF(ARGX.LT.0.) ARGX = 0.
   DO 270 KK=I,MI
      ZZZ = ZZZ + EXP(XOFFX(KK))*EXP(SLOPEX(KK)*ARGX)
   270 CONTINUE
   IF(I.EQ.IMAXC) ZZZ = (0.5 - CFRAC)*ZZZ
   280 CONTINUE
   ZZZ = ZZZ + BKG
   IF(ZZZ.LE.1.) GO TO 290
   YY(I) = ALOG(ZZZ)
290 CONTINUE
   IF(IGAUSS.EQ.0) GO TO 330
   IBEG = IMAXC - 8
   IEND = NPTS-10
   DO 310 I=IBEG,IEND
      GCOR = 0.
      DO 300 K=38,54
         INDX = I+46-K
         IF(INDX.LE.0) GO TO 300
         YDATA = EXP(YY(INDX)) - BKG
         GCOR = GCOR + GAUSS(K)*YDATA
      300 CONTINUE
      Z(I) = GCOR + BKG
      Z(I) = ALOG(Z(I))
   310 CONTINUE
   YY(I) = Z(I)
320 CONTINUE
330 CONTINUE
   STDTMP = 0.
   VARTMP = 0.
   NBEG = IMAXC - 6
   DO 340 I=NBEG,NPTS
      STDTMP = STDTMP + (EXP(YY(I)) - FLOAT(ICNT(I)))**2
   OBSRV = 1.
   IF(ICNT(I).NE.0) OBSRV = 1./FLOAT(ICNT(I))
   VARTMP = VARTMP + OBSRV*(EXP(YY(I)) - FLOAT(ICNT(I)))**2
340 CONTINUE
   STDTMP = SQRT(STDTMP/FLOAT(NPTS-NBEG))
   VARTMP = VARTMP/(FLOAT(NPTS-NBEG+1 - 7))
   IF(NPASS.EQ.1) STDLIN = STDTMP
   IF(NPASS.EQ.1) VARLIN = VARTMP
   IF(NPASS.EQ.2) STDLIN = STDTMP
   IF(NPASS.EQ.2) VARLIN = VARTMP
C+++++
C
C WRITE OUT THE FITTED SPECTRUM IF REQUESTED
C
C+++++
IF(IPRFLG(NPASS+4).EQ.0) GO TO 360
WRITE(6,13)
DO 350 I=1,NPTS
FIT = EXP(YY(I))
WRITE(6,7) DATAX(I),DATAX(I),DATAY(I),FIT
350 CONTINUE
360 CONTINUE
FITMAX = EXP(YY(IMAXC))
RELMAX = DATAY(IMAXC)
RATMAX = RELMAX/FITMAX
IF(NPASS.EQ.1) WRITE(6,15) STDLIN,VARIIN
IF(NPASS.EQ.2) WRITE(6,16) STNDLIN,VAILIN
C+++++
C
C PLOT THE FITTED SPECTRUM IF REQUESTED
C
C+++++
IF(IPLFLG(NPASS+4).EQ.0) GO TO 370
CALL DATPLT(DATAX,DATAYL,4*NPASS,SLOPEX,XOFF,NPL(1),1,NPO(1),MFIT)
370 CONTINUE
IF(NPASS.EQ.2) GO TO 400
C+++++
C
C DETERMINE NONLINEAR FIT
C
C+++++
SOL(1) = EXP(XOFFX(1))
SOL(1) = SOL(1)*RATMAX
SOL(2) = -SLOPEX(1)
SOL(3) = EXP(XOFFX(2))
SOL(3) = SOL(3)*RATMAX
SOL(4) = -SLOPEX(2)
SOL(5) = EXP(XOFFX(3))
SOL(5) = SOL(5)*RATMAX
SOL(6) = -SLOPEX(3)
SOL(7) = 0.5
DO 380 I=1,NPTS
DATAY(I) = DATAY(I) - BKG
380 CONTINUE
SOL1 = SOL(1)
SOL2 = SOL(2)
SOL3 = SOL(3)
SOL4 = SOL(4)
SOL5 = SOL(5)
SOL6 = SOL(6)
SOL7 = SOL(7)
390 CONTINUE
SOL(1) = SOL1
SOL(2) = SOL2
SOL(3) = SOL3
SOL(4) = SOL4
SOL(5) = SOL5
SOL(6) = SOL6
SOL(7) = SOL7
CALL LSQ6(NPTS,X,DATAY,SOL,E,ITER1,IERR,STD,IMAXC,
1 NPO,NPL,CFRAC,ICT)
SOL1 = SOL(1)
SOL2 = SOL(2)
SOL3 = SOL(3)
SOL4 = SOL(4)
SOL5 = SOL(5)
SOL6 = SOL(6)
SOL7 = SOL(7)
SLOPEX(1) = - SOL(2)
SLOPEX(2) = - SOL(4)
SLOPEX(3) = - SOL(6)
ESL1 = E(2)
ESL2 = E(4)
ESL3 = E(6)
EX01 = ALOG(E(1))
EX02 = ALOG(E(3))
EX03 = ALOG(E(5))
SOL(2) = SOL(3)
SOL(3) = SOL(5)
SOL(4) = CFRAC
SOL(5) = -SLOPEX(1)
SOL(6) = -SLOPEX(2)
SOL(7) = -SLOPEX(3)
CFRACO = CFRAC
CALL LSQ1
XOFFX(1) = ALOG(SOL(1))
XOFFX(2) = ALOG(SOL(2))
XOFFX(3) = ALOG(SOL(3))
TZERO = FLOAT(IMAXC) + SOL(4) + 0.5
ETZERO = E(4)
CFRAC = SOL(4)
C+++++
C
C WRITE OUT FINAL FIT IF REQUESTED
C
C+++++
IF(IPRFLG(3).EQ.0) GO TO 250
SLOPE1 = SLOPEX(1)
SLOPE2 = SLOPEX(2)
SLOPE3 = SLOPEX(3)
XOFF1 = XOFFX(1)
XOFF2 = XOFFX(2)
XOFF3 = XOFFX(3)
TAU1 = -1./SLOPE1
TAU1 = TAU1*TSF
TAU2 = -1./SLOPE2
TAU2 = TAU2*TSF
TAU3 = -1./SLOPE3
TAU3 = TAU3*TSF
ETAU1 = ESL1/SLOPE1**2
ETAU1 = ETAU1*TSF
ETAU2 = ESL2/SLOPE2**2
ETAU2 = ETAU2*TSF
ETAU3 = ESL3/SLOPE3**2

16
ETAU3 = ETAU3*TSF
AREA1 = TAU1*EXP(XOFF1)
AREA2 = TAU2*EXP(XOFF2)
AREA3 = TAU3*EXP(XOFF3)
EAREA1 = ETAU1*EXP(EXOI) + EXOI*TAU1
EAREA2 = ETAU2*EXP(EXO2) + EXO2*TAU2
EAREA3 = ETAU3*EXP(EXO3) + EXO3*TAU3
AREAT = AREA1 + AREA2 + AREA3
EAREAT = EAREA1 + EAREA2 + EAREA3
XINTI = AREAI/AREAT
XINT2 = AREA2/AREAT
XINT3 = AREA3/AREAT
EINTI = EAREAI/AREAT + EAREAT*AREAI/AREAT^2
EINT2 = EAREA2/AREAT + EAREAT*AREA2/AREAT^2
EINT3 = EAREA3/AREAT + EAREAT*AREA3/AREAT^2
WRITE(6,14) SLOPE1,ESLOPE1,SLOPE2,ESL2,SLOPE3,ESL3,
1 TAU1,ETAU1,TAU2,ETAU2,TAU3,ETAU3,
2 XOFF1,EXOI,XOFF2,EXO2,XOFF3,EXO3,
3 AREAI,EAREAI,AREA2,EAREA2,AREA3,EAREA3,
4 XINT1,XINT2,XINT3,EINT1,EINT2,EINT3,
5 TZERO,ETZERO,STD,ITER1,ITER2,IERR
DELTO = CFRAC - CFRACO
IF(ICT.NE.0) DELTO = 4.*DELTO
CFRAC = CFRACO + DELTO
IF(ABS(CFRAC - CFRACO).LE.0.001) ICT = ICT + 1
IF(ABS(CFRAC - CFRACO).GT.0.001
1 .OR.ICT.LT.2) GO TO 390
GO TO 250
400 CONTINUE

C-----------------------------------------------------------------------------
C Accurate
C This subroutine is documented in "Langley Graphics System", Central Scientific Computing Complex Document G-3
C-----------------------------------------------------------------------------

CALL CALPLT(0.,0.,999)
STOP
END

SUBROUTINE DATPLT(X,Y,ICODE,SLOPE,XOFF,NPLST,NPOA,NFIT)

COMMON/PRDAT/DATAX(800),DATAY(800),DATAX'(800),DATAYL(800)
COMMON/PEEK/IPEEK(2)
COMMON/OUTDAT/SLOPEX(4),XOFFX(4),ESLOPE(4),EXOFF(4),
1 NPIST(4),NPLST(4),TAU(4),ETAU(4),AREA(4),EAREA(4)
COMMON/INDAT/ICNT(800),NPTS,NPTSS,BKG,NDJV,IMAXC,TSF,IDCHAN,
1 IKPLFLG(10),LPFLGL(10),NPO(4),NPL(4),LABIL,CFRAC
DIMENSION JMESS(4),KMESS(3)
COMMON/FITDAT/XX(800),YY(800)
DIMENSION X(800),Y(800)
DIMENSION MESS(2)
DIMENSION MESSI(4)
DATA IENT/0/
DATA JMESS/IOHFIRST FIT,IOHSECOND FIT,IOHTHIRD FIT,IOHFORTH FIT/
DATA KMESS/IOHOR FIRST,IOHOR SECOND,IOHOR THIRD /
NPSY = NPTS
IF(IPEEK(1).EQ.0) GO TO 20
YMAX = O.
DO 10 I=1,NPST
IF(Y(I).LT.YMAX) GO TO 10
YMAX = Y(I)
IMAX = I
10 CONTINUE
NPST = IPEEK(2) - IPEEK(1) + I
20 CONTINUE
IF(ICODE.EQ.3.OR.ICODE.EQ.5) NPST = NPTSS
IENT = IENT + 1

C---------------------------------------------
C
C SUBROUTINE NFRAME IS A PART OF THE LANGLEY GRAPHICS
C SYSTEM AND EXECUTES A FRAME ADVANCE
C---------------------------------------------

IF(IENT.GT.1) CALL NFRAME

C---------------------------------------------
C
C SUBROUTINE CALPLT IS A PART OF THE LANGLEY GRAPHICS
C SYSTEM AND MOVES THE PEN TO A NEW LOCATION WITH PEN
C UP OR DOWN. CALPLT CAN ALSO ESTABLISH A NEW PLOT
C ORIGIN
C---------------------------------------------

CALL CALPLT(1.5,1.5,-3)
MESS(1) = IOHLN (COUNTS
MESS(2) = IOH)
IF(ICODE.EQ.1) MESSI(1) = IOHRAW SPECTR
IF(ICODE.EQ.1) MESSI(2) = IOHUM
IF(ICODE.EQ.2) MESSI(1) = IOHSPECTRUM M
IF(ICODE.EQ.2) MESSI(2) = IOHINUS BACKG
IF(ICODE.EQ.2) MESSI(3) = IOHROUND
IF(ICODE.EQ.3) MESSI(1) = IOHSPECTRUM C
IF(ICODE.EQ.3) MESSI(2) = IOHORRECTED F
IF(ICODE.EQ.3) MESSI(3) = KMESS(NFIT)
IF(ICODE.EQ.3) MESSI(4) = IOHFIT
IF(ICODE.EQ.4.OR.ICODE.EQ.8) MESSI(1) = IOHFITTED SPE
IF(ICODE.EQ.4.OR.ICODE.EQ.8) MESSI(2) = IOHCTRUM
IF(ICODE.EQ.5) MESSI(1) = JMESS(NFIT)
INDXI = I
IF(IPEEK(1).NE.0) INDXI = IMAX + IPEEK(1)
INDX2 = NPST
IF(IPEEK(1).NE.0) INDX2 = IMAX + IPEEK(2)
YSV = DATAYL(INDX1)
SUBROUTINE ASCALE IS A PART OF THE LANGLEY GRAPHICS SYSTEM AND COMPUTES A SCALING FACTOR FOR AN ARRAY OF DATA. ASCALE ALSO DETERMINES THE DATA MINIMUM

IF(IENT.EQ.1) CALL ASCALE(DATAYL(INDX1),.,NPST,1,10.)
DATAYL(INDX1) = YSV
IF(IENT.EQ.1) YMIN = DATAYL(INDX2+1)
IF(IENT.EQ.1) YSF = DATAYL(INDX2+2)
Y(INDX2+1) = YMIN
Y(INDX2+2) = YSF
ISF = ALOG10(X(NPST)/10.)
SF = 10.**ISF
VP = X(NPST)/SF
IVP = VP
IVP = IVP + 1
VMAX = IVP*SF
VMAX = VMAX/10.
X(NPST+1) = 0.
X(NPST+2) = VMAX
MESS2 = 7HCHANNEL

SUBROUTINE NOTATE IS A PART OF THE LANGLEY GRAPHICS SYSTEM AND DRAWS ALPHANUMERIC INFORMATION

CALL NOTATE(4.5,7.1,0.2,LABEL,0.,7)
IF(ICODE.EQ.5) GO TO 30
IF(ICODE.EQ.4.OR.ICODE.EQ.8) GO TO 50

SUBROUTINE AXES IS A PART OF THE LANGLEY GRAPHICS SYSTEM AND DRAWS A LINE, ANNOTATES THE VALUE OF A VARIABLE AT SPECIFIED INTERVALS WITH TIC MARKS AND PROVIDES AN AXIS IDENTIFICATION LABEL

CALL AXES(0.,0.,0.,10.,X(NPST+1),X(NPST-2),1.,0.,
1 MESS2,0.2,-7)
CALL AXES(0.,0.,90.,8.,Y(INDX2+1),Y(INDX2+2),1.,0.,
1 MESS1,0.2,11)
IF(ICODE.EQ.1) CALL NOTATE(4.5,7.5,0.2,MESS1,0.,12)
IF(ICODE.EQ.2) CALL NOTATE(4.5,7.5,0.2,MESS1,0.,25)
IF(ICODE.EQ.3) CALL NOTATE(4.5,7.5,0.2,MESS1,0.,33)
IF(ICODE.EQ.4.OR.ICODE.EQ.8)
1CALL NOTATE(4.5,7.5,0.2,MESS1,0.,15)
IF(ICODE.EQ.8) CALL NOTATE(3.5,7.5,0.2,FINAL,0.,5)

SUBROUTINE LINPLT IS A PART OF THE LANGLEY GRAPHICS SYSTEM AND DRAWS A LINE BETWEEN AND/OR DRAWS A NASA...
C STANDARD SYMBOL AT EACH SUCCESSIVE DATA POINT IN AN
C ARRAY
C
C-----------------------------------------------------------------------
CALL LINPLT(X,Y(INDX1),NPST,1,-1,22,1,0)
30 CONTINUE
IF(ICODE.LE.4) GO TO 60
NP = NPLA + NPOA
DO 40 I=1,NP
XX(I) = FLOAT(I - 1)
YY(I) = SLOPE*XX(I) + XOFF
40 CONTINUE
XX(NP+1) = 0.
ISF = ALOG10(XX(NP)/10.)
SF = 10.*ISF
VP = XX(NP)/SF
IVP = VP
IVP = IVP + 1
VMAX = IVP*SF
VMAX = VMAX/10.
XX(NP+2) = VMAX
YY(NP+1) = YMIN
YY(NP+2) = YSF
XSV1 = X(NP+1)
XSV2 = X(NP+2)
YSV1 = Y(NP+1)
YSV2 = Y(NP+2)
X(NP+1) = XX(NP+1)
X(NP+2) = XX(NP+2)
Y(NP+1) = YY(NP+1)
Y(NP+2) = YY(NP+2)
CALL AXES(O.,O.,O.,IO.,X(NP+I),X(NP+2),I.,O.,
C MESS2,0.2,-7)
CALL AXES(O.,O.,90.,8.,Y(NP+I),Y(NP+2),I.,O.,
C MESS(1),O.2,11)
CALL NOTATE(4.5,7.5,0.2,MESSI,O.,10)
CALL LINPLT(X,Y,NP,I,-I,22,1,O)
CALL LINPLT(XX,YY,NP,I,O,O,O,O)
XXP = XX(NPOA)/XX(NP+2)
yyp = yy(NPOA)/YY(NP+2)
C-----------------------------------------------------------------------

C SUBROUTINE PNTPLT IS A PART OF THE LANGLEY GRAPHICS
C SYSTEM AND DRAWS A NASA STANDARD SYMBOL CENTERED ON
C A GIVEN COORDINATE
C
C-----------------------------------------------------------------------
CALL PNTPLT(XXP,YYP,3,2)
XXP = XX(NP)/XX(NP+2)
yyp = YY(NP)/YY(NP+2)
CALL PNTPLT(XXP,YYP,3,2)
X(NP+1) = XSV1
X(NP+2) = XSV2
Y(NP+1) = YSV1
Y(NP+2) = YSV2
GO TO 60
50 CONTINUE
   Y(NPST+1) = YMIN
   Y(NPST+2) = YSF
   CALL AXES(0.,0.,0.,0.,X(NPST+1),X(NPST+2),1.,0.,
   1 MESS2,0.2,-7)
   CALL AXES(0.,0.,90.,90.,Y(NPST+1),Y(NPST+2),1.,0.,
   1 MESS1,0.2,11)
   IF(ICODE.EQ.8) CALL NOTATE(3.5,7.5,0.2,5'HFINAL,O.,5)
   CALL NOTATE(4.5,7.5,0.2,MESS1,O.,15)
   CALL LINPLT(X,Y,NPST,I,-I,22,1,O)
   YY(NPST+1) = YMIN
   YY(NPST+2) = YSF
   XX(NPST+1) = X(NPST+1)
   XX(NPST+2) = X(NPST+2)
   CALL LINPLT(XX(1),YY(1),NPST,I,O,O,O(),
   60 CONTINUE
   RETURN
END

SUBROUTINE DATRD

SUBROUTINE DATRD READS THE EXPERIMENTAL DATA AND
THE OPERATION INSTRUCTIONS

COMMON/INDAT/ICNT(800),NPTS,NPTSS,BKG,NDIV,IMAXC,TSF,IDCHAN,
1 IPRFLG(10),IPLFLG(10),NPO(4),NPL(4),ABEL,CFRAC
COMMOM/MXNIT/ITERMX
COMMON/PEEK/IPEEK(2)
COMMON/GAUSDAT/AMP,SIGINV,IGAUSS,GAUS_S(91)
NAMELIST/INDAT/BKG,NDIV,TSF,IDCHAN,NPO,NPL,IPRFLG,IPLFLG,
1 AMP,SIGINV,IGAUSS,IPEEK,ITERMX,CFRAC
1 FORMAT(14,A7)
REWIND 1
READ(1,1) NPTS,LABEL
READ(1,*) (ICNT(I),I=1,NPTS)
BKG = 0.
NDIV = 51
N1 = NPTS - NDIV + 1
DO 10 I=N1,NPTS
   BKG = BKG + FLOAT(ICNT(I))
10 CONTINUE
   BKG = BKG/FLOAT(NDIV)
TSF = 61.57
IDCHAN = -2
NPO(1) = 120
NPL(1) = 30
NPO(2) = 38
NPL(2) = 20
NPO(3) = 11
NPL(3) = 9
NPO(1) = NPO(1) - IDCHAN
NPO(2) = NPO(2) - IDCHAN
NPO(3) = NPO(3) - IDCHAN
NPO(4) = 0
NPL(4) = 0

21
DO 20 I=1,10
IPRFLG(I) = 0
IPLFLG(I) = 1
20 CONTINUE
IPRFLG(3) = 1
IPRFLG(6) = 1
IPLFLG(2) = 0
AMP = 0.22387
SIGINV = 0.15745
IGAUSS = 1
ITERMX = 30
CFRAC = -0.5
IPEEK(1) = 0
IPEEK(2) = 0
READ(5,INDAT,END=30)
30 IF(EOF(5).NE.0) GO TO 40
40 CONTINUE
IF(IGAUSS.EQ.0) AMP = 0.
IF(IGAUSS.NE.2) GO TO 50
READ(1,*) (GAUSS(KK),KK=38,54)
50 CONTINUE
WRITE(6,INDAT)
RETURN
END
SUBROUTINE LSQ2(N,X,Y,SLOPE,XOFF,ITER,IERR,STD,E1,E2)
SUBROUTINE LSQ2 OBTAINS THE LEAST-SQUARE FIT OF
A STRAIGHT LINE TO A PORTION OF THE EXPERIMENTAL
DATA IN THE LOG DOMAIN
DIMENSION X(30),Y(30),ARAY(2),R(100),B(2,3),C(2)
DIMENSION KARY(7),ERROR(2)
DATA ERROR/2*I.E-5/
IERR = 0
XOFF = 0.
SLOPE = 20.
ITER = 0
10 CONTINUE
DO 20 I=1,2
DO 20 J=1,3
B(I,J) = 0.
20 CONTINUE
ITER = ITER + 1
DO 50 I=1,N
ARAY(1) = X(I)
ARAY(2) = 1.
FX = SLOPE*X(I) + XOFF
R(I) = Y(I) - FX
ISUM = 1
DO 40 K=1,2
B(K,ISUM) = B(K,ISUM) + ARAY(ISUM)**2
B(K,3) = B(K,3) + ARAY(K)*R(I)
IF(ISUM.EQ.2) GO TO 40
JM1 = ISUM
ISUM = ISUM + 1
40 CONTINUE
WRITE(6,INDAT)
RETURN
END
DO 30 J=ISUM,2
B(K,J) = B(K,J) + ARAY(JM1)*ARAY(J)
30 CONTINUE
40 CONTINUE
50 CONTINUE
   DO 70 I=1,2
   IM1 = I - 1
   IF(I.EQ.1) GO TO 70
   DO 60 J=1,IM1
      B(I,J) = B(J,I)
60 CONTINUE
70 CONTINUE
KARY(1) = 10
KARY(2) = 2
KARY(3) = 3
KARY(4) = 0
KARY(5) = 2
KARY(6) = 0
KARY(7) = 0

C---------------------------------------------------------------------
C
C SUBROUTINE MATOPS OBTAINS THE INVERSE OF A MATRIX AND THE
C SOLUTION OF A SET OF LINEAR EQUATIONS. THIS SUBROUTINE IS
C DOCUMENTED IN "MATHEMATICAL AND STATISTICAL SOFTWARE AT
C LANGLEY", CENTRAL SCIENTIFIC COMPUTING COMPLEX DOCUMENT
C N2-3A
C
C---------------------------------------------------------------------

CALL MATOPS(KARY,B,DET,DUMMY)
DO 80 I=1,2
   C(I) = B(I,3)
80 CONTINUE
IF(SLOPE.EQ.0.) GO TO 110
TEST = ABS(C(1)/SLOPE)
IF(TEST.GT.ERROR(1)) GO TO 150
90 IF(XOFF.EQ.0.) GO TO 120
TEST = ABS(C(2)/XOFF)
IF(TEST.GT.ERROR(2)) GO TO 150
100 CONTINUE
   GO TO 130
110 IF(ABS(C(1)).GT.ERROR(1)) GO TO 150
   GO TO 90
120 IF(ABS(C(2)).GT.ERROR(2)) GO TO 150
   GO TO 100
130 CONTINUE
   SLOPE = SLOPE + C(1)
   XOFF = XOFF + C(2)
   STD = 0.
   DO 140 IJ=1,N
      STD = STD + R(IJ)**2
140 CONTINUE
   STD = STD/FLOAT(N-2)
   E1 = B(1,1)*STD
   E2 = B(2,2)*STD
   E1 = SQRT(E1)

23
E2 = SQRT(E2)
STD = FLOAT(N-2)*STD
STD = SQRT(STD/FLOAT(N))
RETURN
150 IF(ITER.GT.30) GO TO 160
SLOPE = SLOPE + C(1)
XOFF = XOFF + C(2)
GO TO 10
160 IERR = 1
GO TO 130
END
SUBROUTINE LSQ6(NPTS,X,Y,SOL,E,ITER,IERR,STD,IMAXC,
NPO,NPL,CFRAC,ICT)
SUBROUTINE LSQ6 LOCATES THE AMPLITUDES AND LIFETIMES FOR FIXED ZERO TIME THAT MINIMIZE THE STANDARD DEVIATION BETWEEN THE ANALYTIC APPROXIMATION AND THE EXPERIMENTAL DATA
DIMENSION NPL(4),NPO(4)
DIMENSION FXC(800),DER(6,800)
DIMENSION X(800),Y(800)
COMMON/MXNIT/ITERMX
COMMON/GAUSDAT/AMP,SIGINV,IGAUSS,GAUSS(91)
DIMENSION E(6)
DIMENSION R(800)
DIMENSION SOL(6)
DIMENSION ARAY(25),B(6,7),C(25)
DIMENSION KARY(7),ERROR(25)
DATA ERROR/25*I.E-5/
DATA EPS/I.E-5/
DATA NENTRY/O/
ITER = 0
IERR = 99
E(1) = 1.
E(2) = 1.
E(3) = 1.
E(4) = 1.
E(5) = 1.
E(6) = 1.
NENTR = NENTR + 1
IF(NENTR.EQ.1) RETURN
IERR = 0
IPT = IMAXC + 3
IF(ICT.EQ.1) IPT = IMAXC - 6
ITER = 0
10 CONTINUE
DO 20 I=1,6
DO 20 J=1,7
B(I,J) = 0.
20 CONTINUE
ITER = ITER + 1
NVAR = 6
NXO = IMAXC
XO = FLOAT(IMAXC)
NPTSP = NPTS + 20
DO 30 JJ=1,NPTSP
   IF(JJ.LT.NXO-16) GO TO 30
   ARGX = FLOAT(JJ) - (XO + CFRAC)
   IF(JJ.EQ.NXO) ARGX = 0.5*(0.5 - CFRAC)
   IF(ARGX.LT.0.) ARGX = 0.
   ARG = SOL(2)*ARGX
   C1 = 0.
   IF(ARG.LT.670.) C1 = EXP(-ARG)
   DER(1,JJ) = C1
   DER(2,JJ) = -SOL(1)*ARGX*DER(1,JJ)
   ARG = SOL(4)*ARGX
   C2 = 0.
   IF(ARG.LT.670.) C2 = EXP(-ARG)
   DER(3,JJ) = C2
   DER(4,JJ) = -SOL(3)*ARGX*DER(3,JJ)
   ARG = SOL(6)*ARGX
   C3 = 0.
   IF(ARG.LT.670.) C3 = EXP(-ARG)
   DER(5,JJ) = C3
   DER(6,JJ) = -SOL(5)*ARGX*DER(5,JJ)
   FXC(JJ) = SOL(1)*DER(1,JJ) + SOL(3)*DER(3,JJ) + SOL(5)*DER(5,JJ)
   IF(JJ.EQ.NXO) FXC(JJ) = (0.5 - CFRAC)*FXC(JJ)
   IF(JJ.LT.NXO) FXC(JJ) = 0.
   IF(JJ.LT.NXO) DER(1,JJ) = 0.
   IF(JJ.LT.NXO) DER(3,JJ) = 0.
   IF(JJ.LT.NXO) DER(5,JJ) = 0.
30 CONTINUE
   DO 80 I=1,NPTS
      R(1) = 0.
      FX = 0.
      DO 40 JJ=1,6
         ARAY(JJ) = 0.
      40 CONTINUE
      IF(I.LT.IPT) GO TO 80
      NST = I-8
      NEND = I + 8
      KK = 37
      DO 50 JJ=NST,NEND
         KK=KK+I
         ARAY(1) = ARAY(1) + DER(1,JJ)*GAUSS(KK)
         ARAY(2) = ARAY(2) + DER(2,JJ)*GAUSS(KK)
         ARAY(3) = ARAY(3) + DER(3,JJ)*GAUSS(KK)
         ARAY(4) = ARAY(4) + DER(4,JJ)*GAUSS(KK)
         ARAY(5) = ARAY(5) + DER(5,JJ)*GAUSS(KK)
         ARAY(6) = ARAY(6) + DER(6,JJ)*GAUSS(KK)
         FX = FX + FXC(JJ)*GAUSS(KK)
50 CONTINUE
   R(I) = Y(I) - FX
   ISUM = 1
   DO 70 K=1,NVAR
      B(K,ISUM) = B(K,ISUM) + ARAY(ISUM)**2
      B(K,NVAR+1) = B(K,NVAR+1) + ARAY(K)*R(I)
   70 CONTINUE
   IF(ISUM.EQ.NVAR) GO TO 70
   JM1 = ISUM
C
C ISUM = ISUM + 1
DO 60 J=ISUM,NVAR
B(K,J) = B(K,J) + ARAY(JM1)*ARAY(J)
60 CONTINUE
70 CONTINUE
80 CONTINUE
DO 100 I=1,NVAR
IM1 = I - 1
IF(I.EQ.1) GO TO 100
DO 90 J=1,IM1
B(I,J) = B(J,I)
90 CONTINUE
100 CONTINUE
KARY(1) = 10
KARY(2) = 6
KARY(3) = 7
KARY(4) = 0
KARY(5) = NVAR
KARY(6) = 0
KARY(7) = 0

C-------------------------------------------------------------
C SUBROUTINE MATOPS OBTAINS THE INVERSE OF A MATRIX AND THE
C SOLUTION OF A SET OF LINEAR EQUATIONS. THIS SUBROUTINE IS
C DOCUMENTED IN "MATHEMATICAL AND STATISTICAL SOFTWARE AT
C LANGLEY", CENTRAL SCIENTIFIC COMPUTING COMPLEX DOCUMENT
C N2-3A
C-------------------------------------------------------------
CALL MATOPS(KARY,B,DET,DUMMY)
DO 110 I=1,NVAR
C(I) = B(I,NVAR+1)
110 CONTINUE
DO 120 K=1,NVAR
TST = ABS(C(K))
IF(ABS(SOL(K)).GT.EPS) TST = ABS(C(K)/SOL(K))
IF(TST.GT.ERROR(K)) GO TO 170
120 CONTINUE
DO 130 IJ=1,NVAR
SOL(IJ) = SOL(IJ) + C(IJ)
130 CONTINUE
140 CONTINUE
STD = 0.
DO 150 IJ=1,NPTS
IF(IJ.LT.IPT) GO TO 150
STD = STD + R(IJ)**2
150 CONTINUE
STD = STD/FLOAT(NPTS-NXO-2)
DO 160 IJ=1,NVAR
E(IJ) = SQRT(ABS(B(IJ,IJ))*STD)
160 CONTINUE
STD = FLOAT(NPTS-NXO-2)*STD
SrO = SQRT(SrD/FLOAT(NPTS-NXO))
RETURN
170 IF(ITER.GT.ITERMX) GO TO 200
RATMAX = 0.
DO 180 IJ=1,NVAR
RAT = ABS(C(IJ)/SOL(IJ))
IF(RAT.GT.RATMAX) RATMAX = RAT
180 CONTINUE
FRAC = 1.
IF(RATMAX.GT.0.4) FRAC = 0.4/RATMAX
DO 190 IJ=1,NVAR
SOL(IJ) = SOL(IJ) + FRAC*C(IJ)
190 CONTINUE
FRAC = I.
IF(RATMAX.GT.0.4) FRAC = 0.4/RATMAX
DO 200 IJ=1,NVAR
SOL(IJ) = SOL(IJ) + FRAC*C(IJ)
200 IERR = 1
GO TO 140
END

SUBROUTINE LSQI

SUBROUTINE LSQI LOCATES THE VALUE OF ZERO TIME FOR FIXED AMPLITUDES AND LIFETIMES WHICH MINIMIZES THE STANDARD DEVIATION BETWEEN THE ANALYTIC APPROXIMATION AND THE EXPERIMENTAL DATA.

COMMON/M_XNIT/ITERMX
COMMON/FUNCOM/R(800),XO,NXO
COMMON/GAUSDAT/AMP,SIGINV,IGAUSS,GAUSS(9)
DIMENSION E(7)
DIMENSION SOL(7)
COMMON/NEWCOM/SOL,E,ITER,IERR,STD
COMMON/PDAT/DATAX(800),DATAY(800),DATAXT(800),DATAYL(800)
COMMON/CLCDAT/X(800),Y(800),YL(800)
COMMON/INDAT/ICNT(800),NPTS,NPTSS,BKG,NDIV,IMAXC,TSF,IDCHAN,
1 IPRFLG(IO),IPLFLG(IO),NPO(4),NPL(4),LABEL,CFRAC
DATA EPS/I.E-5/
DATA NENTRY/O/
NENTRY = NENTRY + 1
IERR = 0
ITER = 0
NXO = IMAXC
XO = FLOAT(IMAXC)
CALL FUNX(STD0,SOL)
DELX = 0.05
IF(NENTRY.GT.1) DELX = 0.01
SOL(4) = SOL(4) + DELX
CALL FUNX(STD,SOL)
IF(STD.GE.STDO) DELX = -DELX
STDO = STD
SOL(4) = SOL(4) + DELX
10 CONTINUE
ITER = ITER + 1
CALL FUNX(STD,SOL)
IF(STD.GE.STDO) GO TO 20
STD0 = STD
SOL(4) = SOL(4) + DELX
GO TO 10
20 TST = ABS(STD0 - STD)/STD0
IF(TST.LT.EPS) GO TO 30
SOL(4) = SOL(4) - 2.0*DELX
DELX = 0.5*DELX
STDO = 1.E99
GO TO 10
30 CONTINUE
STD = 0.
DO 40 IJ=1,NPTS
IF(IJ.LT.NXO) GO TO 40
STD = STD + R(IJ)**2
40 CONTINUE
STD = STD/FLOAT(NPTS-NXO-2)
E(4) = ABS(DELX)
RETURN
END

SUBROUTINE FUNX(STD,SOL)
SUBROUTINE FUNX CALCULATES THE STANDARD DEVIATION
BETWEEN THE ANALYTIC APPROXIMATION AND THE EXPERIMENTAL
DATA. THE THREE AMPLITUDES ARE IN SOL(1), SOL(2),
AND SOL(3). THE ZERO TIME IS IN SOL(4). THE THREE
LAMBDAS (I/TAU) ARE IN SOL(5), SOL(6), AND SOL(7).

COMMON/FUNCOM/R(800),XO,NXO
COMMON/PRDAT/DATAX(800),DATAY(800),DATAXT(800),DATAYL(800)
DIMENSION SOL(7)
COMMON/GAUSDAT/AMP,SIGINV,IGAUSS,GAUSS(91)
COMMON/INDAT/ICNT(800),NPTS,NPTSS,BKG,NDIV,IMAXC,TSF,ICCHAN,
1 IPRFLG(IO),IPLFLG(IO),NPO(4),NFL(4),LABEL,CFRAC
DIMENSION CONT(800)
NPTSP = NPTS + 20
DO 10 JJ=1,NPTSP
IF(JJ.LT.NXO-20) GO TO 10
ARG = SOL(5)*(FLOAT(JJ) - (XO + SOL(4)))
IF(JJ.EQ.NXO) ARG = SOL(5)*0.5*(0.5 - SOL(4))
C1 = 0.
IF(ARG.LT.670.) C1 = EXP(-ARG)
ARG = SOL(6)*(FLOAT(JJ) - (XO + SOL(4)))
IF(JJ.EQ.NXO) ARG = SOL(6)*0.5*(0.5 - SOL(4))
C2 = 0.
IF(ARG.LT.670.) C2 = EXP(-ARG)
ARG = SOL(7)*(FLOAT(JJ) - (XO + SOL(4)))
IF(JJ.EQ.NXO) ARG = SOL(7)*0.5*(0.5 - SOL(4))
C3 = 0.
IF(ARG.LT.670.) C3 = EXP(-ARG)
CONT(JJ) = SOL(1)*C1 + SOL(2)*C2 + SOL(3)*C3
IF(JJ.EQ.NXO) CONT(JJ) = (0.5 - SOL(4))*CONT(JJ)
IF(JJ.LT.NXO) CONT(JJ) = 0.
10 CONTINUE
DO 30 I=1,NPTS
FX = 0.
IF(I.LT.NXO-8) GO TO 30
NST = I-8
NEND = I + 8
KK = 37
DO 20 JJ=NST,NEND

28
KK = KK + 1
FX = FX + CONT(JJ) * GAUSS(KK)
20 CONTINUE
   R(I) = DATAY(I) - FX
30 CONTINUE
   STD = 0.
   DO 40 IJ = 1, NPTS
      IF(IJ .LT. NXO - 8) GO TO 40
      STD = STD + R(IJ)**2
   40 CONTINUE
   RETURN
END
0 0 0 0 0
0 0 1 4 0
1 2 5 4 1
3 3 4 6 7
7 5 7 3 10
6 5 6 7 11
11 9 5 9 13
6 5 10 6 6
11 11 2 8 11
18 23 12 12 17
14 14 12 5 12
15 12 18 14 20
11 17 12 10 9
16 15 17 8 13
14 16 16 16 16
12 18 13 11 13
12 16 9 12 14
23 14 16 15 13
24 13 15 20 24
104 379 1627 5549 17239
42056 80350 122509 152716 160806
152636 134497 115315 97223 82346
69670 60471 51668 42047 39372
34291 30716 27085 24209 21504
19336 17210 15327 14182 12738
11623 10827 9787 9075 8404
7740 7123 6843 6381 5945
5570 5427 5157 4921 4531

TOTAL COUNTS = 1872471
NUMBER OF CHANNELS IN DATA FILE = 650
BACKGROUND AVG. = 1.29020E+01  NDIV = 51
NUMBER OF COMPONENTS TO FIT = 3

Figure A1. Input spectrum.
<table>
<thead>
<tr>
<th>CHANNEL</th>
<th>TIME (PS)</th>
<th>COUNTS</th>
<th>LOG(COUNTS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>6.15700E+01</td>
<td>000000E+00</td>
<td>000000E+00</td>
</tr>
<tr>
<td>2.</td>
<td>1.23140E+02</td>
<td>000000E+00</td>
<td>000000E+00</td>
</tr>
<tr>
<td>3.</td>
<td>1.84170E+02</td>
<td>000000E+00</td>
<td>000000E+00</td>
</tr>
<tr>
<td>4.</td>
<td>2.46280E+02</td>
<td>000000E+00</td>
<td>000000E+00</td>
</tr>
<tr>
<td>5.</td>
<td>3.07650E+02</td>
<td>000000E+00</td>
<td>000000E+00</td>
</tr>
<tr>
<td>6.</td>
<td>3.69420E+02</td>
<td>000000E+00</td>
<td>000000E+00</td>
</tr>
<tr>
<td>7.</td>
<td>4.30990E+02</td>
<td>000000E+00</td>
<td>000000E+00</td>
</tr>
<tr>
<td>8.</td>
<td>4.92560E+02</td>
<td>1000000E+00</td>
<td>1.38629E+00</td>
</tr>
<tr>
<td>9.</td>
<td>5.54130E+02</td>
<td>4000000E+00</td>
<td>1.93147E-01</td>
</tr>
<tr>
<td>10.</td>
<td>6.15700E+02</td>
<td>000000E+00</td>
<td>000000E+00</td>
</tr>
<tr>
<td>11.</td>
<td>6.77270E+02</td>
<td>1000000E+00</td>
<td>1.38629E+00</td>
</tr>
<tr>
<td>12.</td>
<td>7.38440E+02</td>
<td>2000000E+00</td>
<td>1.38629E+00</td>
</tr>
<tr>
<td>13.</td>
<td>8.00410E+02</td>
<td>5000000E+00</td>
<td>1.38629E+00</td>
</tr>
<tr>
<td>14.</td>
<td>8.61980E+02</td>
<td>4000000E+00</td>
<td>1.38629E+00</td>
</tr>
<tr>
<td>15.</td>
<td>9.23550E+02</td>
<td>1000000E+00</td>
<td>000000E+00</td>
</tr>
<tr>
<td>16.</td>
<td>9.85120E+02</td>
<td>3000000E+00</td>
<td>1.09861E+00</td>
</tr>
<tr>
<td>17.</td>
<td>1.04669E+03</td>
<td>3000000E+00</td>
<td>1.09861E+00</td>
</tr>
<tr>
<td>18.</td>
<td>1.10826E+03</td>
<td>4000000E+00</td>
<td>1.38629E+00</td>
</tr>
<tr>
<td>19.</td>
<td>1.16982E+03</td>
<td>6000000E+00</td>
<td>1.79170E+00</td>
</tr>
<tr>
<td>20.</td>
<td>1.23140E+03</td>
<td>7000000E+00</td>
<td>1.94591E+00</td>
</tr>
<tr>
<td>21.</td>
<td>1.29297E+03</td>
<td>7000000E+00</td>
<td>1.94591E+00</td>
</tr>
<tr>
<td>22.</td>
<td>1.35454E+03</td>
<td>5000000E+00</td>
<td>1.60944E+00</td>
</tr>
<tr>
<td>23.</td>
<td>1.41611E+03</td>
<td>7000000E+00</td>
<td>1.94591E+00</td>
</tr>
<tr>
<td>24.</td>
<td>1.47763E+03</td>
<td>3000000E+00</td>
<td>1.09861E+00</td>
</tr>
<tr>
<td>25.</td>
<td>1.53925E+03</td>
<td>1000000E+01</td>
<td>2.30259E+00</td>
</tr>
<tr>
<td>26.</td>
<td>1.60082E+03</td>
<td>8000000E+00</td>
<td>1.79170E+00</td>
</tr>
<tr>
<td>27.</td>
<td>1.66239E+03</td>
<td>5000000E+00</td>
<td>1.60944E+00</td>
</tr>
<tr>
<td>28.</td>
<td>1.72396E+03</td>
<td>5000000E+00</td>
<td>1.79170E+00</td>
</tr>
<tr>
<td>29.</td>
<td>1.78553E+03</td>
<td>7000000E+00</td>
<td>1.94591E+00</td>
</tr>
<tr>
<td>30.</td>
<td>1.84710E+03</td>
<td>1000000E+01</td>
<td>2.3979UE+00</td>
</tr>
<tr>
<td>31.</td>
<td>1.90867E+03</td>
<td>1000000E+00</td>
<td>2.3979UE+00</td>
</tr>
<tr>
<td>32.</td>
<td>1.97024E+03</td>
<td>7000000E+00</td>
<td>2.19726E+00</td>
</tr>
</tbody>
</table>

Figure A2. Sample raw spectrum output.
SPECTRUM MAXIMUM = $1.60853 \times 10^5$

CHANNEL NUMBER CORRESPONDING TO SPECTRUM MAXIMUM = 105

CORRECTED CHANNEL = 103

Figure A3. Sample peak output.

FIRST FIT (FROM 137 TO 167)

SLOPE = $-3.54090816 \times 10^{-2}$ (3.11094633E-03)

TAU = 1.73584868E+03 (1.52540054E+02)

INTERCEPT = 9.46815652E+00 (4.69875567E-01)

AREA = 2.24636328E+07 (1.97483780E+06)

STANDARD DEVIATION = 1.42757049E-01

ITERATIONS = 2

IERR = 0

Figure A4. Sample fit output (initial estimate).
### FITTED SPECTRUM

**CHANNEL** | **TIME (PS)** | **COUNTS** | **FIT**
--- | --- | --- | ---
97 | 5.97229E+03 | 3.79000E+02 | 2.00127E+02
98 | 6.03368E+03 | 1.62700E+03 | 1.12727E+03
99 | 6.09543E+03 | 5.59000E+03 | 4.94784E+03
100 | 6.15700E+03 | 1.72390E+04 | 1.60466E+04
101 | 6.2157E+03 | 4.20560E+04 | 4.13228E+04
102 | 6.28014E+03 | 8.03600E+04 | 8.02394E+04
103 | 6.34171E+03 | 1.22509E+05 | 1.22844E+05
104 | 6.40328E+03 | 1.52710E+05 | 1.53026E+05
105 | 6.46483E+03 | 1.60660E+05 | 1.61660E+05
106 | 6.52642E+03 | 1.52630E+05 | 1.52150E+05
107 | 6.58799E+03 | 1.34497E+05 | 1.33940E+05
108 | 6.64956E+03 | 1.15315E+05 | 1.14537E+05
109 | 6.71113E+03 | 9.72300E+04 | 9.72127E+04
110 | 6.77270E+03 | 5.23460E+04 | 5.26300E+04
111 | 6.83427E+03 | 6.96700E+04 | 7.05280E+04
112 | 6.89584E+03 | 4.04710E+04 | 4.04880E+04
113 | 6.95741E+03 | 5.16680E+04 | 5.23160E+04
114 | 7.01898E+03 | 4.50370E+04 | 4.51060E+04
115 | 7.08055E+03 | 3.93720E+04 | 3.93270E+04
116 | 7.14212E+03 | 3.42910E+04 | 3.44189E+04
117 | 7.20369E+03 | 3.07160E+04 | 3.02750E+04
118 | 7.26526E+03 | 2.70850E+04 | 2.67640E+04
119 | 7.32683E+03 | 2.42090E+04 | 2.37760E+04
120 | 7.38840E+03 | 2.15040E+04 | 2.12230E+04
121 | 7.44997E+03 | 1.93360E+04 | 1.90330E+04
122 | 7.51154E+03 | 1.72100E+04 | 1.71450E+04
123 | 7.57311E+03 | 1.53270E+04 | 1.55110E+04
124 | 7.63468E+03 | 1.41820E+04 | 1.40900E+04
125 | 7.69625E+03 | 1.27260E+04 | 1.28400E+04
126 | 7.75782E+03 | 1.16230E+04 | 1.17630E+04
127 | 7.81939E+03 | 1.08270E+04 | 1.08060E+04
128 | 7.88096E+03 | 9.78700E+03 | 9.60170E+03
129 | 7.94253E+03 | 9.07500E+03 | 9.20410E+03
130 | 8.00410E+03 | 8.40400E+03 | 8.53980E+03

**NONLINEAR FIT STANDARD DEVIATION:** $1.05934364E+02$

**VARIANCE:** $1.80950085E+00$

Figure A5. Sample fitted spectrum output (initial estimate).
NONLINEAR FIT

SLOPE = -3.556485E-02 (± 1.044433E-03)  
TAU = 1.731204E+03 (± 2.108365E+01)  
INTERCEPT = 9.509106E+00 (± 6.353532E+00)  
AREA = 2.133997E+07 (± 6.451731E+04)  
INTENSITY = 2.034002E-01 (± 1.345535E-02)  
TO = 1.031930E+02 (± 7.812500E-05)  

STANDARD DEVIATION = 7.703474E+03  
ITERATIONS = 9,  30  
ICT = 1  
IERR = 0

Figure A6. Sample results from final fit.
Figure A7. Sample graphic output.
Appendix B

Least-Squares Solution

The solution of equation (A1) consists of seven parameters: $A_1$, $A_2$, $A_3$, $\lambda_1$, $\lambda_2$, $\lambda_3$, and $t_z$, where $A_i$ is the amplitude of the $i$th component, $t_z$ is time zero, $\lambda_i = 1/\tau_i$, and $\tau_i$ is the lifetime of the $i$th component. The method of least squares has been used to obtain the solution. This method is extremely sensitive to the initial estimated values of the seven parameters; accordingly, a multistep procedure is employed to obtain crude initial estimates and then refine them. This procedure is described in appendix A.

In general, equation (A1) can be written as follows:

$$n_k = F_k(t_j,A_1,A_2,A_3,\lambda_1,\lambda_2,\lambda_3) + \varepsilon_k \tag{B1}$$

where the measurement error in channel $k$ is given by $\varepsilon_k$. Expanding equation (B1) in a Taylor series and dropping higher order terms yields the following linear approximation:

$$n_k = (n_k)_0 + \frac{\partial F_k}{\partial A_1} (A_1 - A_1,0) + \frac{\partial F_k}{\partial A_2} (A_2 - A_2,0) + \frac{\partial F_k}{\partial A_3} (A_3 - A_3,0) + \frac{\partial F_k}{\partial \lambda_1} (\lambda_1 - \lambda_1,0) + \frac{\partial F_k}{\partial \lambda_2} (\lambda_2 - \lambda_2,0) + \frac{\partial F_k}{\partial \lambda_3} (\lambda_3 - \lambda_3,0) + \varepsilon_k \tag{B2}$$

Equation (B2) can be simplified with the following definitions:

$$b_{k,1} = \frac{\partial F_k}{\partial A_1}, \quad b_{k,2} = \frac{\partial F_k}{\partial A_2}, \quad b_{k,3} = \frac{\partial F_k}{\partial A_3}, \quad b_{k,4} = \frac{\partial F_k}{\partial \lambda_1}, \quad b_{k,5} = \frac{\partial F_k}{\partial \lambda_2}, \quad b_{k,6} = \frac{\partial F_k}{\partial \lambda_3}$$

$$\Delta\alpha_1 = A_1 - A_1,0 \quad \Delta\alpha_2 = A_2 - A_2,0 \quad \Delta\alpha_3 = A_3 - A_3,0 \quad \Delta\alpha_4 = \lambda_1 - \lambda_1,0 \quad \Delta\alpha_5 = \lambda_2 - \lambda_2,0 \quad \Delta\alpha_6 = \lambda_3 - \lambda_3,0$$

The simplified equation (B2) takes the form:

$$\Delta n_k = \sum_{l=1}^{6} b_{k,l} \Delta\alpha_l + \varepsilon_k \tag{B3}$$

or, in matrix notation,

$$Z_k = B_k \Delta\alpha + \varepsilon_k \tag{B4}$$

where

$$Z_k = \Delta n_k,$$

$$\Delta\alpha = \begin{bmatrix} \Delta\alpha_1 \\ \Delta\alpha_2 \\ \vdots \\ \Delta\alpha_6 \end{bmatrix},$$

$$B_k = [b_{k,1}, b_{k,2}, b_{k,3}, \ldots, b_{k,6}],$$

and

$$e_k = [\varepsilon_k]$$

Thus, for $N$ channels, there are $N$ matrix equations, which may be written as

$$\bar{Z} = \bar{B} \Delta\alpha + \bar{\varepsilon} \tag{B5}$$

where

$$\bar{Z} = \begin{bmatrix} Z_1 \\ Z_2 \\ \vdots \\ Z_N \end{bmatrix}, \quad \bar{B} = \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_N \end{bmatrix}, \quad \bar{\varepsilon} = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_N \end{bmatrix}$$

The sum of the squares of the residuals

$$\bar{\varepsilon}^T \bar{\varepsilon} = (\bar{Z} - \bar{B} \Delta\alpha)^T (\bar{Z} - \bar{B} \Delta\alpha) \tag{B6}$$

is a minimum for that value of $\Delta\alpha$ for which the first variation with respect to $\Delta\alpha$ of equation (B6) vanishes if the second variation with respect to $\Delta\alpha$ is positive. Both these conditions are satisfied for

$$\Delta\alpha' = (\bar{B}^T \bar{B})^{-1} \bar{B}^T \bar{Z} \tag{B7}$$

The value of $\Delta\alpha$ obtained from equation (B7) is used to determine new values of $A_i$ and $\lambda_i$, and this iteration process continues until $\Delta\alpha' \rightarrow 0$. This process leads to the best values of $A_i$ and $\lambda_i$.

The derivatives included in equation (B2) are readily determined from equation (A1) and are given
by the following expressions:

\[
\frac{\partial F_k}{\partial A_1} = \sum_{j=0}^{\infty} W_{j,k} e^{-\lambda_1(t_j - t_2)}
\]

\[
\frac{\partial F_k}{\partial A_2} = \sum_{j=0}^{\infty} W_{j,k} e^{-\lambda_2(t_j - t_2)}
\]

\[
\frac{\partial F_k}{\partial A_3} = \sum_{j=0}^{\infty} W_{j,k} e^{-\lambda_3(t_j - t_2)}
\]

\[
\frac{\partial F_k}{\partial \lambda_1} = -\sum_{j=0}^{\infty} W_{j,k}(t_j - t_2) A_1 e^{-\lambda_1(t_j - t_2)}
\]

\[
\frac{\partial F_k}{\partial \lambda_2} = -\sum_{j=0}^{\infty} W_{j,k}(t_j - t_2) A_2 e^{-\lambda_2(t_j - t_2)}
\]

\[
\frac{\partial F_k}{\partial \lambda_3} = -\sum_{j=0}^{\infty} W_{j,k}(t_j - t_2) A_3 e^{-\lambda_3(t_j - t_2)}
\]
Appendix C

Positron Lifetime Measurements in Epoxy Samples

As part of an ongoing study for elucidating the role of metal ions in controlling free volume in polymers, the following chromium compounds were introduced in Epon 828 epoxy resin. The metal-complex concentration in each case was 1 for every 10 repeat units of the host epoxy.

1. Chromic acetate [Cr(Ac)₃]
2. Chromic perchlorate [Cr(DMSO)₆ (ClO₄)₃]
3. Chromous chloride [CrCl₂]

The samples were prepared in the form of 1-in-diameter by 0.1-in-thick discs. Positron lifetime spectra were measured in these samples by using a standard fast-fast coincidence measurement system. (See refs. 12 to 15.) The experimental lifetime spectra are listed in table CI. These spectra have been analyzed using the program for analyzing positron lifetime spectra (PAPLS) and POSITRONFIT program. The results of these analyses have been summarized in tables I to III in the text.
Table CI. Experimental Lifetime Spectra Observed in Epon 828 Epoxy Sample Containing 0.1 Mole Fraction of Transition Metal Compounds

(a) Cr(Ac)$_3$

<table>
<thead>
<tr>
<th>Sample</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>5</td>
<td>4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>5</td>
<td>2</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>9</td>
<td>7</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>12</td>
<td>8</td>
<td>6</td>
<td>7</td>
<td>10</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>12</td>
<td>7</td>
<td>11</td>
<td>16</td>
<td>5</td>
<td>11</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>9</td>
<td>6</td>
<td>10</td>
<td>10</td>
<td>5</td>
<td>7</td>
<td>9</td>
</tr>
<tr>
<td>6</td>
<td>11</td>
<td>9</td>
<td>9</td>
<td>16</td>
<td>9</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>10</td>
<td>11</td>
<td>7</td>
<td>5</td>
<td>8</td>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td>5</td>
<td>12</td>
<td>9</td>
<td>6</td>
<td>14</td>
<td>9</td>
<td>16</td>
<td>22</td>
</tr>
<tr>
<td>31</td>
<td>235</td>
<td>839</td>
<td>3467</td>
<td>11143</td>
<td>25965</td>
<td>48945</td>
<td>73446</td>
</tr>
<tr>
<td>91292</td>
<td>95538</td>
<td>90162</td>
<td>78976</td>
<td>67947</td>
<td>57669</td>
<td>49002</td>
<td>41565</td>
</tr>
<tr>
<td>35696</td>
<td>30604</td>
<td>26904</td>
<td>23265</td>
<td>20652</td>
<td>18178</td>
<td>15907</td>
<td>14229</td>
</tr>
<tr>
<td>12556</td>
<td>11396</td>
<td>10186</td>
<td>9117</td>
<td>8407</td>
<td>7534</td>
<td>6821</td>
<td>6286</td>
</tr>
<tr>
<td>5811</td>
<td>5439</td>
<td>4998</td>
<td>4814</td>
<td>4314</td>
<td>3947</td>
<td>3753</td>
<td>3587</td>
</tr>
<tr>
<td>3525</td>
<td>3121</td>
<td>3131</td>
<td>2968</td>
<td>2706</td>
<td>2619</td>
<td>2528</td>
<td>2328</td>
</tr>
<tr>
<td>2375</td>
<td>2126</td>
<td>2079</td>
<td>2027</td>
<td>1915</td>
<td>1787</td>
<td>1698</td>
<td>1573</td>
</tr>
<tr>
<td>1626</td>
<td>1470</td>
<td>1406</td>
<td>1311</td>
<td>1325</td>
<td>1247</td>
<td>1142</td>
<td>1160</td>
</tr>
<tr>
<td>1104</td>
<td>1091</td>
<td>1042</td>
<td>927</td>
<td>963</td>
<td>932</td>
<td>859</td>
<td>847</td>
</tr>
<tr>
<td>801</td>
<td>783</td>
<td>761</td>
<td>736</td>
<td>690</td>
<td>694</td>
<td>639</td>
<td>628</td>
</tr>
<tr>
<td>620</td>
<td>611</td>
<td>555</td>
<td>531</td>
<td>534</td>
<td>471</td>
<td>485</td>
<td>493</td>
</tr>
<tr>
<td>497</td>
<td>406</td>
<td>413</td>
<td>424</td>
<td>410</td>
<td>390</td>
<td>405</td>
<td>390</td>
</tr>
<tr>
<td>345</td>
<td>308</td>
<td>327</td>
<td>303</td>
<td>293</td>
<td>273</td>
<td>294</td>
<td>264</td>
</tr>
<tr>
<td>259</td>
<td>248</td>
<td>237</td>
<td>214</td>
<td>220</td>
<td>227</td>
<td>180</td>
<td>205</td>
</tr>
<tr>
<td>218</td>
<td>177</td>
<td>187</td>
<td>184</td>
<td>180</td>
<td>191</td>
<td>172</td>
<td>158</td>
</tr>
<tr>
<td>156</td>
<td>141</td>
<td>138</td>
<td>127</td>
<td>137</td>
<td>110</td>
<td>140</td>
<td>131</td>
</tr>
<tr>
<td>103</td>
<td>115</td>
<td>116</td>
<td>93</td>
<td>113</td>
<td>94</td>
<td>95</td>
<td>95</td>
</tr>
<tr>
<td>88</td>
<td>68</td>
<td>86</td>
<td>71</td>
<td>80</td>
<td>84</td>
<td>75</td>
<td>67</td>
</tr>
<tr>
<td>61</td>
<td>61</td>
<td>73</td>
<td>53</td>
<td>61</td>
<td>57</td>
<td>67</td>
<td>54</td>
</tr>
<tr>
<td>58</td>
<td>55</td>
<td>47</td>
<td>46</td>
<td>51</td>
<td>30</td>
<td>43</td>
<td>50</td>
</tr>
<tr>
<td>36</td>
<td>35</td>
<td>39</td>
<td>52</td>
<td>41</td>
<td>29</td>
<td>31</td>
<td>29</td>
</tr>
<tr>
<td>36</td>
<td>38</td>
<td>36</td>
<td>31</td>
<td>20</td>
<td>29</td>
<td>31</td>
<td>28</td>
</tr>
<tr>
<td>17</td>
<td>24</td>
<td>28</td>
<td>35</td>
<td>38</td>
<td>22</td>
<td>22</td>
<td>24</td>
</tr>
<tr>
<td>15</td>
<td>23</td>
<td>22</td>
<td>28</td>
<td>25</td>
<td>26</td>
<td>23</td>
<td>19</td>
</tr>
<tr>
<td>23</td>
<td>23</td>
<td>21</td>
<td>17</td>
<td>19</td>
<td>22</td>
<td>19</td>
<td>26</td>
</tr>
<tr>
<td>18</td>
<td>10</td>
<td>21</td>
<td>20</td>
<td>18</td>
<td>11</td>
<td>26</td>
<td>21</td>
</tr>
<tr>
<td>19</td>
<td>12</td>
<td>15</td>
<td>11</td>
<td>16</td>
<td>12</td>
<td>14</td>
<td>17</td>
</tr>
<tr>
<td>16</td>
<td>15</td>
<td>14</td>
<td>13</td>
<td>13</td>
<td>10</td>
<td>16</td>
<td>6</td>
</tr>
<tr>
<td>13</td>
<td>18</td>
<td>20</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>12</td>
<td>17</td>
</tr>
<tr>
<td>8</td>
<td>11</td>
<td>12</td>
<td>7</td>
<td>13</td>
<td>10</td>
<td>11</td>
<td>17</td>
</tr>
<tr>
<td>10</td>
<td>13</td>
<td>11</td>
<td>10</td>
<td>11</td>
<td>9</td>
<td>8</td>
<td>13</td>
</tr>
<tr>
<td>19</td>
<td>10</td>
<td>5</td>
<td>7</td>
<td>10</td>
<td>9</td>
<td>11</td>
<td>5</td>
</tr>
<tr>
<td>9</td>
<td>8</td>
<td>13</td>
<td>11</td>
<td>4</td>
<td>9</td>
<td>3</td>
<td>10</td>
</tr>
<tr>
<td>11</td>
<td>9</td>
<td>11</td>
<td>9</td>
<td>8</td>
<td>6</td>
<td>8</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>13</td>
<td>9</td>
<td>9</td>
<td>7</td>
<td>9</td>
<td>12</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>6</td>
<td>11</td>
<td>6</td>
<td>12</td>
<td>8</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>8</td>
<td>10</td>
<td>19</td>
<td>5</td>
<td>9</td>
<td>5</td>
</tr>
<tr>
<td>8</td>
<td>10</td>
<td>11</td>
<td>13</td>
<td>9</td>
<td>9</td>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>14</td>
<td>5</td>
<td>10</td>
<td>9</td>
<td>8</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>9</td>
<td>10</td>
<td>5</td>
<td>7</td>
<td>10</td>
<td>8</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>9</td>
<td>6</td>
<td>8</td>
<td>5</td>
<td>10</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>9</td>
<td>2</td>
<td>6</td>
<td>12</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>10</td>
<td>8</td>
<td>10</td>
<td>10</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>9</td>
<td>6</td>
<td>11</td>
<td>9</td>
<td>6</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td>5</td>
<td>9</td>
<td>9</td>
<td>6</td>
<td>10</td>
<td>9</td>
</tr>
<tr>
<td>7</td>
<td>6</td>
<td>13</td>
<td>13</td>
<td>7</td>
<td>7</td>
<td>9</td>
<td>14</td>
</tr>
<tr>
<td>7</td>
<td>12</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>7</td>
<td>8</td>
<td>12</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>11</td>
<td>7</td>
<td>10</td>
<td>5</td>
<td>11</td>
<td>10</td>
</tr>
</tbody>
</table>
Table CI. Continued

(b) Cr(DMSO)$_6$ (ClO$_4$)$_3$

<table>
<thead>
<tr>
<th></th>
<th>85364</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>6</td>
<td>8</td>
<td>6</td>
<td>5</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>3</td>
<td>6</td>
<td>5</td>
<td>6</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>7</td>
<td>1</td>
<td>8</td>
<td>4</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>5</td>
<td>10</td>
<td>7</td>
<td>5</td>
<td>8</td>
<td>13</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>8</td>
<td>9</td>
<td>11</td>
<td>7</td>
<td>9</td>
<td>9</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>11</td>
<td>4</td>
<td>7</td>
<td>9</td>
<td>9</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>7</td>
<td>8</td>
<td>15</td>
<td>4</td>
<td>12</td>
<td>11</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>14</td>
<td>8</td>
<td>7</td>
<td>6</td>
<td>10</td>
<td>7</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>6</td>
<td>14</td>
<td>7</td>
<td>10</td>
<td>13</td>
<td>15</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>77</th>
<th>242</th>
<th>972</th>
<th>3619</th>
<th>11029</th>
<th>26962</th>
<th>51125</th>
<th>77254</th>
</tr>
</thead>
<tbody>
<tr>
<td>96094</td>
<td>102257</td>
<td>96753</td>
<td>85425</td>
<td>73481</td>
<td>61416</td>
<td>52216</td>
<td>43964</td>
</tr>
<tr>
<td>38232</td>
<td>32262</td>
<td>28756</td>
<td>24449</td>
<td>21391</td>
<td>18693</td>
<td>16812</td>
<td>14781</td>
</tr>
<tr>
<td>12941</td>
<td>11659</td>
<td>10281</td>
<td>9238</td>
<td>8202</td>
<td>7381</td>
<td>6814</td>
<td>6101</td>
</tr>
<tr>
<td>5642</td>
<td>5161</td>
<td>4800</td>
<td>4389</td>
<td>3904</td>
<td>3744</td>
<td>3501</td>
<td>3278</td>
</tr>
<tr>
<td>3208</td>
<td>2863</td>
<td>2807</td>
<td>2635</td>
<td>2566</td>
<td>2332</td>
<td>2222</td>
<td>2084</td>
</tr>
<tr>
<td>1932</td>
<td>1954</td>
<td>1848</td>
<td>1728</td>
<td>1645</td>
<td>1588</td>
<td>1463</td>
<td>1380</td>
</tr>
<tr>
<td>1377</td>
<td>1277</td>
<td>1253</td>
<td>1186</td>
<td>1141</td>
<td>1099</td>
<td>1077</td>
<td>980</td>
</tr>
<tr>
<td>955</td>
<td>958</td>
<td>890</td>
<td>897</td>
<td>825</td>
<td>814</td>
<td>748</td>
<td>721</td>
</tr>
<tr>
<td>683</td>
<td>711</td>
<td>656</td>
<td>639</td>
<td>629</td>
<td>592</td>
<td>561</td>
<td>536</td>
</tr>
<tr>
<td>564</td>
<td>479</td>
<td>501</td>
<td>456</td>
<td>451</td>
<td>469</td>
<td>427</td>
<td>391</td>
</tr>
<tr>
<td>410</td>
<td>380</td>
<td>395</td>
<td>362</td>
<td>340</td>
<td>333</td>
<td>310</td>
<td>321</td>
</tr>
<tr>
<td>315</td>
<td>295</td>
<td>269</td>
<td>250</td>
<td>240</td>
<td>243</td>
<td>231</td>
<td>257</td>
</tr>
<tr>
<td>202</td>
<td>204</td>
<td>181</td>
<td>179</td>
<td>201</td>
<td>182</td>
<td>182</td>
<td>183</td>
</tr>
<tr>
<td>175</td>
<td>145</td>
<td>163</td>
<td>164</td>
<td>143</td>
<td>142</td>
<td>131</td>
<td>130</td>
</tr>
<tr>
<td>114</td>
<td>123</td>
<td>107</td>
<td>115</td>
<td>86</td>
<td>99</td>
<td>109</td>
<td>114</td>
</tr>
<tr>
<td>107</td>
<td>110</td>
<td>95</td>
<td>89</td>
<td>96</td>
<td>78</td>
<td>98</td>
<td>78</td>
</tr>
<tr>
<td>68</td>
<td>75</td>
<td>75</td>
<td>67</td>
<td>75</td>
<td>68</td>
<td>52</td>
<td>61</td>
</tr>
<tr>
<td>52</td>
<td>58</td>
<td>49</td>
<td>58</td>
<td>49</td>
<td>47</td>
<td>44</td>
<td>48</td>
</tr>
<tr>
<td>50</td>
<td>57</td>
<td>45</td>
<td>37</td>
<td>40</td>
<td>37</td>
<td>40</td>
<td>33</td>
</tr>
<tr>
<td>33</td>
<td>31</td>
<td>33</td>
<td>36</td>
<td>25</td>
<td>25</td>
<td>33</td>
<td>25</td>
</tr>
<tr>
<td>34</td>
<td>38</td>
<td>27</td>
<td>31</td>
<td>30</td>
<td>23</td>
<td>33</td>
<td>29</td>
</tr>
<tr>
<td>19</td>
<td>25</td>
<td>14</td>
<td>23</td>
<td>27</td>
<td>25</td>
<td>21</td>
<td>24</td>
</tr>
<tr>
<td>19</td>
<td>18</td>
<td>20</td>
<td>34</td>
<td>22</td>
<td>22</td>
<td>23</td>
<td>14</td>
</tr>
<tr>
<td>26</td>
<td>27</td>
<td>16</td>
<td>26</td>
<td>24</td>
<td>22</td>
<td>11</td>
<td>9</td>
</tr>
<tr>
<td>16</td>
<td>14</td>
<td>16</td>
<td>8</td>
<td>19</td>
<td>15</td>
<td>18</td>
<td>17</td>
</tr>
<tr>
<td>11</td>
<td>9</td>
<td>15</td>
<td>10</td>
<td>19</td>
<td>14</td>
<td>13</td>
<td>14</td>
</tr>
<tr>
<td>10</td>
<td>11</td>
<td>15</td>
<td>14</td>
<td>10</td>
<td>11</td>
<td>8</td>
<td>11</td>
</tr>
<tr>
<td>14</td>
<td>11</td>
<td>8</td>
<td>14</td>
<td>10</td>
<td>5</td>
<td>18</td>
<td>12</td>
</tr>
<tr>
<td>10</td>
<td>12</td>
<td>10</td>
<td>13</td>
<td>12</td>
<td>9</td>
<td>12</td>
<td>10</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>12</td>
<td>14</td>
<td>9</td>
<td>6</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>17</td>
<td>9</td>
<td>13</td>
<td>11</td>
<td>10</td>
<td>11</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>9</td>
<td>7</td>
<td>5</td>
<td>10</td>
<td>6</td>
<td>8</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>7</td>
<td>8</td>
<td>4</td>
<td>11</td>
<td>9</td>
<td>7</td>
</tr>
<tr>
<td>6</td>
<td>8</td>
<td>10</td>
<td>9</td>
<td>12</td>
<td>11</td>
<td>10</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>9</td>
<td>13</td>
<td>14</td>
<td>7</td>
<td>7</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>9</td>
<td>10</td>
<td>3</td>
<td>6</td>
<td>13</td>
<td>6</td>
<td>11</td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td>10</td>
<td>8</td>
<td>7</td>
<td>4</td>
<td>9</td>
<td>12</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>5</td>
<td>13</td>
<td>12</td>
<td>12</td>
<td>16</td>
<td>5</td>
</tr>
<tr>
<td>14</td>
<td>10</td>
<td>4</td>
<td>2</td>
<td>9</td>
<td>9</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>5</td>
<td>8</td>
<td>11</td>
<td>15</td>
<td>9</td>
<td>6</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td>13</td>
<td>11</td>
<td>13</td>
<td>11</td>
<td>9</td>
<td>4</td>
</tr>
<tr>
<td>11</td>
<td>10</td>
<td>9</td>
<td>7</td>
<td>14</td>
<td>7</td>
<td>15</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>3</td>
<td>10</td>
<td>8</td>
<td>5</td>
<td>13</td>
<td>12</td>
<td>13</td>
</tr>
<tr>
<td>12</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>5</td>
<td>10</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>6</td>
<td>7</td>
<td>10</td>
<td>6</td>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td>12</td>
<td>4</td>
<td>9</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td>11</td>
<td>10</td>
<td>6</td>
<td>9</td>
<td>8</td>
<td>7</td>
<td>6</td>
<td>10</td>
</tr>
</tbody>
</table>
### Table CI. C\(\text{r}^{\text{c}}\)luded

#### (c) CrCl\(_2\)

<table>
<thead>
<tr>
<th></th>
<th>4</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>5</th>
<th>4</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>4</td>
<td>6</td>
<td>7</td>
<td>7</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>6</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>3</td>
<td>9</td>
<td>9</td>
<td>13</td>
<td>6</td>
<td>5</td>
<td>10</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>11</td>
<td>11</td>
<td>2</td>
<td>8</td>
<td>11</td>
<td>18</td>
<td>23</td>
</tr>
<tr>
<td>5</td>
<td>12</td>
<td>17</td>
<td>14</td>
<td>14</td>
<td>12</td>
<td>5</td>
<td>12</td>
</tr>
<tr>
<td>6</td>
<td>9</td>
<td>16</td>
<td>15</td>
<td>17</td>
<td>8</td>
<td>13</td>
<td>14</td>
</tr>
<tr>
<td>7</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>12</td>
<td>18</td>
<td>13</td>
<td>11</td>
</tr>
<tr>
<td>8</td>
<td>12</td>
<td>16</td>
<td>9</td>
<td>12</td>
<td>14</td>
<td>23</td>
<td>14</td>
</tr>
<tr>
<td>9</td>
<td>15</td>
<td>13</td>
<td>24</td>
<td>13</td>
<td>15</td>
<td>20</td>
<td>24</td>
</tr>
</tbody>
</table>

| 104| 379 | 1627| 5599| 1'239| 42056| 80360| 122509 |
| 1527| 1608| 665| 1526| 36| 1344| 9722| 82396 |
| 60471| 51668| 45037| 39372| 3391| 30716| 27085| 24209 |
| 21504| 1933| 17210| 15327| 1'482| 12738| 11623| 10827 |
| 9787 | 9075| 8404| 7740| 1'123| 6843| 6381| 5945 |
| 5570 | 5427| 5157| 4921| 1'531| 4413| 4099| 4007 |
| 4005 | 3580| 3537| 3224| 1'189| 3069| 2805| 2734 |
| 2578 | 2370| 2378| 2249| 2'236| 2076| 1994| 1937 |
| 1840 | 1755| 1721| 1641| 1'641| 1560| 1439| 1407 |
| 1360 | 1322| 1262| 1187| 1'158| 1157| 1112| 1072 |
| 1023 | 971 | 1088| 885 | 915 | 873 | 827 | 785 |
| 777  | 684 | 712 | 686 | 648 | 677 | 617 | 597 |
| 581  | 518 | 515 | 507 | 526 | 489 | 452 | 406 |
| 411  | 406 | 412 | 399 | 390 | 381 | 308 | 351 |
| 318  | 311 | 297 | 286 | 269 | 286 | 251 | 236 |
| 219  | 238 | 227 | 232 | 203 | 229 | 214 | 196 |
| 205  | 175 | 165 | 189 | 147 | 156 | 188 | 134 |
| 160  | 135 | 145 | 135 | 121 | 125 | 127 | 118 |
| 103  | 102 | 106 | 108 | 104 | 89  | 111 | 98  |
| 101  | 90  | 73  | 86  | 79  | 69  | 80  | 67  |
| 65   | 79  | 80  | 61  | 84  | 57  | 50  | 67  |
| 52   | 62  | 45  | 55  | 49  | 50  | 64  | 57  |
| 46   | 50  | 48  | 46  | 42  | 39  | 39  | 42  |
| 38   | 40  | 36  | 28  | 38  | 31  | 25  | 29  |
| 34   | 37  | 24  | 27  | 35  | 25  | 28  | 33  |
| 25   | 22  | 31  | 30  | 22  | 36  | 30  | 19  |
| 32   | 21  | 25  | 28  | 22  | 24  | 28  | 16  |
| 22   | 23  | 21  | 24  | 19  | 24  | 15  | 29  |
| 15   | 24  | 19  | 15  | 21  | 24  | 28  | 24  |
| 27   | 16  | 18  | 17  | 20  | 24  | 19  | 20  |
| 16   | 19  | 22  | 20  | 21  | 13  | 18  | 18  |
| 22   | 20  | 16  | 16  | 14  | 20  | 19  | 16  |
| 11   | 15  | 15  | 13  | 12  | 22  | 16  | 19  |
| 14   | 21  | 16  | 11  | 18  | 26  | 14  | 17  |
| 13   | 16  | 9   | 10  | 9   | 17  | 18  | 14  |
| 16   | 11  | 21  | 19  | 27  | 23  | 24  | 15  |
| 14   | 15  | 16  | 9   | 23  | 14  | 12  | 17  |
| 12   | 17  | 12  | 12  | 14  | 8   | 18  | 13  |
| 21   | 16  | 15  | 24  | 17  | 15  | 16  | 12  |
| 11   | 7   | 21  | 10  | 19  | 8   | 17  | 10  |
| 11   | 16  | 11  | 13  | 9   | 22  | 14  | 14  |
| 13   | 16  | 15  | 11  | 14  | 11  | 11  | 17  |
| 19   | 11  | 14  | 15  | 21  | 9   | 8   | 14  |
| 15   | 16  | 15  | 10  | 10  | 17  | 9   | 19  |
| 9    | 9   | 9   | 15  | 16  | 13  | 11  | 13  |
| 15   | 13  | 14  | 20  | 10  | 15  | 17  | 13  |
| 16   | 17  | 18  | 15  | 8   | 14  | 11  | 9   |
| 13   | 14  | 13  | 10  | 10  | 11  | 15  | 41 |
References


Table I. Summary of Input Parameters

\[
\begin{align*}
\tau_i &= \text{Lifetime of } i\text{th component} \\
I_i &= \text{Intensity of } i\text{th component} \\
t_z &= \text{Time zero}
\end{align*}
\]

<table>
<thead>
<tr>
<th>Case 1</th>
<th>Case 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau_1 = 300$ psec; $I_1 = 70%$</td>
<td>$\tau_1 = 300$ psec; $I_1 = 85%$</td>
</tr>
<tr>
<td>$\tau_2 = 700$ psec; $I_2 = 15%$</td>
<td>$\tau_2 = 700$ psec; $I_2 = 10%$</td>
</tr>
<tr>
<td>$\tau_3 = 2100$ psec; $I_3 = 15%$</td>
<td>$\tau_3 = 2100$ psec; $I_3 = 5%$</td>
</tr>
</tbody>
</table>

$t_z = \text{Channel 50.300}$
Background counts = 10.00

Table II. Summary of Results Obtained for Case 1

\[
\begin{align*}
\tau_i &= \text{Lifetime of } i\text{th component} \\
I_i &= \text{Intensity of } i\text{th component} \\
t_z &= \text{Time zero}
\end{align*}
\]

<table>
<thead>
<tr>
<th>PAPLS results</th>
<th>POSITRONFIT results</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau_1 = 298 \pm 1$ psec; $I_1 = 70.49 \pm 0.02%$</td>
<td>$\tau_1 = 310 \pm 2$ psec; $I_1 = 73.80 \pm 0.89%$</td>
</tr>
<tr>
<td>$\tau_2 = 672 \pm 16$ psec; $I_2 = 15.14 \pm 0.02%$</td>
<td>$\tau_2 = 771 \pm 39$ psec; $I_2 = 12.39 \pm 0.67%$</td>
</tr>
<tr>
<td>$\tau_3 = 2080 \pm 22$ psec; $I_3 = 14.37 \pm 0.02%$</td>
<td>$\tau_3 = 2128 \pm 19$ psec; $I_3 = 13.81 \pm 0.33%$</td>
</tr>
</tbody>
</table>

$t_z = \text{Channel 50.320}$
Background counts = 10.27
Standard deviation = 27.46

$t_z = \text{Channel 50.285}$
Background counts = 9.89
Standard deviation = 85.06

Table III. Summary of Results Obtained for Case 2

\[
\begin{align*}
\tau_i &= \text{Lifetime of } i\text{th component} \\
I_i &= \text{Intensity of } i\text{th component} \\
t_z &= \text{Time zero}
\end{align*}
\]

<table>
<thead>
<tr>
<th>PAPLS results</th>
<th>POSITRONFIT results</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau_1 = 298 \pm 1$ psec; $I_1 = 84.44 \pm 0.04%$</td>
<td>$\tau_1 = 308 \pm 1$ psec; $I_1 = 87.64 \pm 0.64%$</td>
</tr>
<tr>
<td>$\tau_2 = 643 \pm 23$ psec; $I_2 = 10.52 \pm 0.03%$</td>
<td>$\tau_2 = 778 \pm 42$ psec; $I_2 = 7.95 \pm 0.49%$</td>
</tr>
<tr>
<td>$\tau_3 = 2021 \pm 59$ psec; $I_3 = 5.04 \pm 0.02%$</td>
<td>$\tau_3 = 2144 \pm 41$ psec; $I_3 = 4.45 \pm 0.22%$</td>
</tr>
</tbody>
</table>

$t_z = \text{Channel 50.321}$
Background counts = 10.27
Standard deviation = 27.64

$t_z = \text{Channel 50.288}$
Background counts = 9.87
Standard deviation = 98.97
Table IV. Comparison of POSITRONFIT and PAPLS Programs for Lifetime Spectra Analysis

[Target system: Epon 828 + Me²⁺; \( \tau_i \) = Lifetime of \( i \)th component; 
\( I_i \) = Intensity of \( i \)th component; \( \sigma \) = Standard deviation; \( t_z \) = Time zero]

<table>
<thead>
<tr>
<th>Target system</th>
<th>POSITRONFIT</th>
<th>PAPLS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( \tau_1 ) (psec)</td>
<td>( I_1 ) (%)</td>
</tr>
<tr>
<td>Epon 828</td>
<td>289±6</td>
<td>52.1±3.2</td>
</tr>
<tr>
<td>containing 0.1 mole fraction of Cr(Ac)₃</td>
<td>Average background = 12.72 counts/channel</td>
<td>( t_z ) = Channel 103.17</td>
</tr>
<tr>
<td>Epon 828</td>
<td>302±8</td>
<td>56.7±4.9</td>
</tr>
<tr>
<td>containing 0.1 mole fraction of Cr(DMSO)₆ (ClO₄)₃</td>
<td>Average background = 7.0 counts/channel</td>
<td>( t_z ) = Channel 103.16</td>
</tr>
<tr>
<td>Epon 828</td>
<td>314±5</td>
<td>63.8±2.1</td>
</tr>
<tr>
<td>containing 0.1 mole fraction of Cr(Cl₂)</td>
<td>Average background = 6.93 counts/channel</td>
<td>( t_z ) = Channel 103.11</td>
</tr>
</tbody>
</table>
Figure 1. Computer-generated spectrum with background counts added.
Figure 2. Computer-generated spectrum that shows effect of finite resolution of counting system.
Figure 3. Typical positron lifetime spectrum in an epoxy target.
(b) Spectrum resolved into three components; $t_z =$ time zero.

Figure 3. Concluded.
Channel number (1 channel = 61.23 psec)

(a) Experimental spectrum, minus background, showing third component; $t_z$ = time zero.

Figure 4. Analysis procedure for a positron lifetime spectrum.
(b) Experimental spectrum, minus background and third component, showing second component clearly.

Figure 4. Continued.
(c) Experimental spectrum showing second component fit.

Figure 4. Continued.
(d) Experimental spectrum minus background, and second and third components.

Figure 4. Continued.
(e) Residual spectrum showing first component fit.

Figure 4. Continued.
Figure 4. Continued.

(f) Comparison of experimental and initial computed spectra.
(g) Comparison of experimental and final computed spectra.

Figure 4. Concluded.
Figure 5. Comparison of PAPLS and POSITRONFIT programs for case 1.
Figure 5. Concluded.
A new procedure for analyzing multicomponent positron lifetime spectra in polymers has been developed. It requires initial estimates of the lifetimes and intensities of various components, which are readily obtainable by a standard spectrum stripping process. These initial estimates, after convolution with the timing-system-resolution function, are then used as the inputs for a nonlinear least-squares analysis to compute the estimates that conform to a global-error minimization criterion. The convolution integral uses the full experimental resolution function, in contrast to the previous studies in which analytical approximations of it were utilized. These concepts have been incorporated into a generalized computer program for analyzing positron lifetime spectra (PAPLS) in polymers. The validity of this program has been tested by using several artificially generated data sets. These data sets were also analyzed with the widely used POSITRONFIT program. In almost all cases, the PAPLS program gives closer fit to the input values. The new procedure has been applied to the analysis of several lifetime spectra measured in metal-ion containing Epon 828 samples. The results are described in this report.