PICOSECOND PHASE FLUOROMETRY

By

ROBERT THOMAS KROUTIL

Bachelor of Science

Oklahoma State University

Stillwater, Oklahoma

1977

Submitted to the Faculty of the Graduate College
of the Oklahoma State University
in partial fulfillment of the requirements
for the Degree of
DOCTOR OF PHILOSOPHY
December, 1982

PICOSECOND PHASE FLUOROMETRY

Thesis Approved:

M. G. Rockley
Thesis Adviser

M. J. Folk

Clarence M. Cunningham

I. Dwaine Eubanks

John A. Gelder

Norman N. Durham
Dean of the Graduate College

ACKNOWLEDGMENTS

The author would like to thank Dr. M. G. Rockley, his thesis adviser, for his many helpful suggestions during the preparation of this thesis. Also, thanks go to Drs. M. J. Folk, C. M. Cunningham, I. D. Eubanks, and J. I. Gelder for serving on his Committee.

Special thanks go to Ms. Janet Sallee for all her help in typing this thesis.

TABLE OF CONTENTS

Chapter		Page
I.	PICOSECOND PHASE FLUOROMETRY.	1
	Methods-History.	1
II.	THE SINGLE PHOTON COUNTING METHOD	36
	1,6 DPH Fluorescent Probe in Lecithin Vesicles . . .	40
III.	THEORETICAL TREATMENT OF PHASE FLUOROMETRY.	46
IV.	EXPERIMENTAL RESULTS.	65
V.	CONCLUSIONS	88
BIBLIOGRAPHY.		91
APPENDIX A - BASIC LANGUAGE PHASE ANGLE SHIFT PROGRAMS.		94
APPENDIX B - PROGRAMS FOR THE SINGLE PHOTON COUNTING METHOD . . .		158

LIST OF TABLES

Table		Page
I.	Deconvolution by the Phase Plane Method.	13
II.	Kerr Constants of Some Selected Liquids.	28
III.	Absorbance of DPH at 360 NM.	42
IV.	DPH Lifetimes.	44
V.	%RSD Versus Phase Modulation Frequencies	48
VI.	Tangent Function Errors.	51
VII.	% Relative Error Vs. Well Depth.	54
VIII.	% Relative Error Vs. Well Depth Lifetime	57
IX.	Simulation Curves-10 K Max. Counts Well Depth.	59
X.	Simulation Curves-100 K Max. Counts Well Depth	61
XI.	Phase Shift Computer Programs Actual Lifetime = 1.3912 nsec	68
XII.	Experimental Lifetimes (0.8441 nsec)	70
XIII.	8 K Well Depth	71
XIV.	Warm-Up Period of the Photomultiplier.	73
XV.	Degree Shift Vs. % Error	76
XVI.	Experimental Delay Line Measurements (10 K Counts) . . .	81
XVII.	Lifetime Shift	85
XVIII.	Average Lifetimes.	86
XIX.	Description and Explanation Steps.	153
XX.	Execution Time	155

LIST OF FIGURES

Figure — Page

1. Free-Running Lamp. 6

2. Variable Delay Timing. 9

3. The Phase Fluorometer of Gaviola 19

4. Phase Shift Versus Modulation Measurements 21

5. Phase and Modulation Depth Measurement 26

6. Single Photon Counting Apparatus 37

7. DPH Incorporation of Lipid Vesicles. 43

8. Percent Relative Standard Deviation Versus the Fluorescent Lifetime Measurement - 10K Count Maximum 60

9. Percent Relative Standard Deviation Versus the Fluorescent Lifetime Measurement - 100K Count Maximum. 62

10. Optical Diagram. 67

11. Phase Shift Lifetime Versus Real Run Time. 74

12. Experimental Percent Relative Standard Deviation Versus the Phase Angle Measurement. 77

13. Lifetime Measurement Versus the Photomultiplier Photon Count Rate 79

14. Rate Process Diagram 83

CHAPTER I

PICOSECOND PHASE FLUOROMETRY

Methods-History

The decay pathways of electronic excited states of molecules can be classified as radiative, non-radiative, or chemical processes. From an electronic ground state a molecule can undergo an excitation into a higher excited electronic state from which several non-radiative decay processes can occur. These can be classified as internal conversion, intersystem crossing, isomerization, and decomposition.

Internal conversion is the process by which the molecule relaxes from a higher excited electronic state (S_1) to a lower state (S_o) by a non-radiative mechanism (1). This transition is between electronic levels of different energies with the same spin multiplicity. Intersystem crossing is that process in which a molecule has crossed from the singlet excited state to a triplet excited state. That is, intersystem crossing refers to that process by which the spin multiplicity of the excited state is altered. The excited triplet state can then undergo intersystem crossing to the ground singlet state via a radiative step, phosphorescence, or by a non-radiative process. The triplet state has a long lifetime because the intersystem crossing process is spin-forbidden (2).

When a molecule undergoes excitation to a higher excited energy level its quantum mechanical wave function is a mixing of the initial wavefunction times a time dependent perturbation coefficient (4).

$$\psi(x,t) = \Sigma a_k(t)\psi_{(k)} \tag{1}$$

In other words, the initial state evolves in time into other wavefunctions. Kauzmann (5) shows that the probability of the coefficient $a_f(t)$ is just,

$$[a_f(t)]^2 = \frac{8\pi^3}{3h^2} <\psi_i|\mu|\psi_f>^2 \, p(V_{1f})t \tag{2}$$

where $<\psi|\mu|\psi>$ is the transition moment, $p(V_{1f})$ is the radiation density per unit volume, t is time, and μ is the dipole moment (3). The absorption and emission of radiation are usually given in terms of the Einstein coefficients. The coefficient of absorption is defined as the square of the time dependent coefficient per unit volume.

$$B_{1u} = \frac{8\pi<\psi_1|\mu|\psi_\mu>^2}{3h^2} \tag{3}$$

Likewise, the coefficient of stimulated emission is equal to the coefficient of absorption. That is,

$$B_{1u} = B_{u1} \tag{4}$$

where B_{u1} is the coefficient of transition from state u to the ground state and B_{1u} is the probability of absorption from the ground state to u. In order to account for the fact that an inequality of population occurs in the upper versus lower electronic state, a third coefficient must be introduced. This constant, commonly known as the spontaneous emission coefficient, A_{ul}, is independent of the radiation density. It can be shown that the spontaneous emission coefficient can be related to the stimulated emission coefficient by the expression, (1)

$$A_{ul} = \frac{64\pi^4 \nu^3}{3hc^3} \langle \psi_u | \mu | \psi_l \rangle^2 \tag{5}$$

Therefore the rate of spontaneous emission is directly related to the square of the transition moment and the cube of the transition frequency. The spontaneous rate constant can be evaluated if the population of the electronically excited species is known relative to the ground state.

The spontaneous rate coefficient can be evaluated by showing that if a molecule A is excited by an ideal impulse and relaxes into the ground state, then the concentration of A* (the excited state of A) can be expressed as,

$$\frac{d[A^*]}{dt} = -k'[A^*] + f(t) \tag{6}$$

where,

$$A \overset{k''}{\rightleftharpoons} [A^*],\ f(t) = k''[A] \cong \text{constant for small changes in } [A] \tag{7}$$

and,

$$k' = \sum_i k_i \tag{8}$$

where k' is the sum of the rate constants for the several modes of decay.

If the exciting function is an infinitely narrow exciting pulse, then f(t) equals zero and Equation (1) reduces to:

$$\frac{d[A^*]}{[A^*]} = -\Sigma k dt \qquad (9)$$

or,

$$A^* = \exp\{-\Sigma kt\} \qquad (10)$$

The constant k is usually expressed in terms of the fluorescence lifetime. The fluorescence lifetime is related to the reciprocal of the sum of the decay rate constants and is defined as,

$$\tau_F = \frac{1}{\sum_i k_i}$$

For organic substances most typical fluorescence lifetimes for electronic transitions of the type S_1 to S_o are 10^{-8} to 10^{-11} seconds. The measured value is related to the fluorescence quantum yield by,

$$\Phi = \tau_F/\tau_o$$

In this equation Φ is the fluorescence quantum yield and τ_o is equal to the mean natural radiative lifetime (1). That is,

$$\tau_o = \tau_F \cdot k_R$$

where k_R is the radiative rate constant. The quantum yield is then defined by the number of molecules undergoing a process divided by the number of quanta of light absorbed. Only when the fluorescence quantum yield is unity will the measured lifetime value equal the mean natural radiative lifetime.

Fluorescent lifetimes can be measured by several different methods.

Among these are the phase shift method and the pulse method. The first pulse fluorometer used for the direct observation of lifetimes was built by Brody in 1957 (6). The pulse fluorometer is based on the use of a pulsed light source. The source used was a nitrogen-filled free-running gas lamp.

Nitrogen-filled gas lamps are essentially an RC timing circuit. In a typical free running lamp a discharge takes place between two closely spaced electrodes at the breakdown voltage. Changes in the gas pressure, electrode geometry, and voltage applied alter the breakdown frequency of the lamp and the intensity of the resulting flash. In most applications the voltage applied to the lamps is between 5 and 20 KV (7) with the resistance on the order of megohms and the lamps running at a pressure in the range of 0.1 to 100 atmospheres. The free-running lamp shown in Figure 1 has a trigger pick-off to indicate lamp pulse firing. This trigger pick-off pulse indicates the firing rate of the lamp and can be used to synchronize subsequent experiments or measurements.

An alternative lamp that can be employed is the hydrogen thyratron lamp (8, 56). The thyratron is a tube filled with hydrogen gas to which a voltage is applied. The applied voltage enables the tube to gate a pulse at the positive grid. The breakdown voltage applied from the thyratron is sent to a pair of platinum electrodes. The voltage is usually several times the breakdown voltage. Thus, once the thyratron is discharged the lamp will discharge.

Munro (1) improved the design of the pulse fluorometer using a sampling oscilloscope instead of a traveling wave oscilloscope. The sampling oscilloscope gave improved signal-to-noise ratios, enabling better time resolution and sensitivity. The advantage obtained by Munro

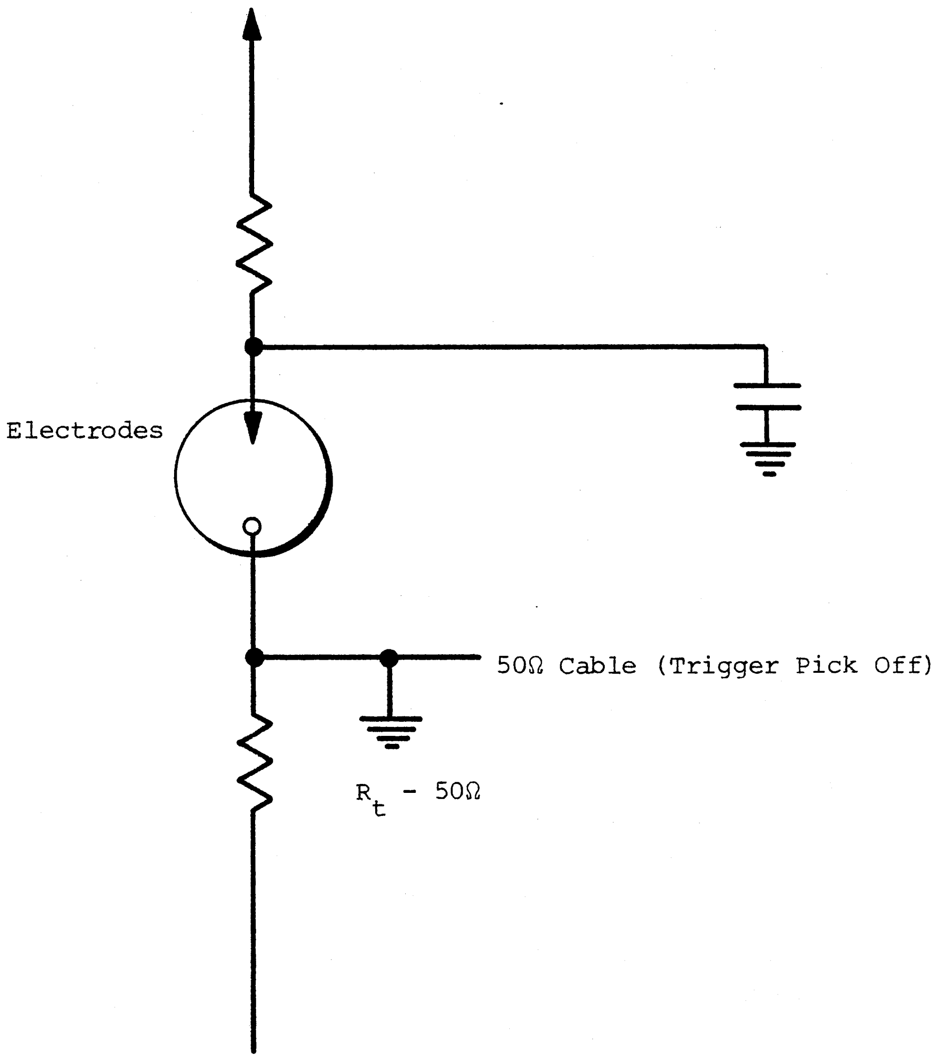

Figure 1. Free-Running Lamp

came from being able to integrate the intensity of the signal electronically at any point. A plot for further detailed analysis was then obtained on a recorder.

Another measurement technique is the "stroboscopic" method first introduced by Bennett (8). This technique involves sampling the lifetime through a photomultiplier tube which is configured as a traveling wave amplifier. The photomultiplier gate is synchronized with the lamp flash and the sampling is done from a time base which has a variable delay timing circuit to provide the time sweep. The variable delay timing circuit increasingly delays the photomultiplier turn-on time with respect to the pulse pick-off time from the lamp. An output which is linearly proportional to the number of photons received is sent to a recorder. The advantage of this technique over the design of Munro (1) is that in Munro's instrument the response time was determined by the transit time spread of the electron avalanche through the photomultiplier while in Bennett's instrument the photomultiplier is switched on for a much shorter time than the sample decay time. Using an improved photomultiplier tube, Bennett pulsed the tube dynodes in succession rather than simultaneously. The stroboscopic technique of pulsing the cathode works by setting up a voltage gradient between adjacent elements of the tube. When a negative-going portion of a ramp pulse is received at the cathode, a voltage gain between the elements produces a multiplication of the electrons cascading through the tube. When the pulse is flat or negative, no multiplication of electrons occurs. The stroboscopic method eliminates the transit time spread of the electron avalanche because only those electrons that leave the cathode in phase with the voltage ramp are collected. The time resolution achieved by Bennett was

1.8 nsec. The output was typically a plot of the photocurrent versus the delay time between the lamp gate and the photomultiplier turn on time.

The most recent development in the use of flash techniques has been the single-photon-counting time-of-flight method. This technique possesses better sensitivity and reliability than earlier pulsed instruments because the performance is independent of lamp intensity and frequency stability and involves sensitivities down to the single photon counting range. A measure of the success of this technique is indicated by the fact that the instrumentation is now commercially available.

Each time the lamp is flashed, a voltage sweep is started in a time-to-amplitude converter or TAC. The voltage of the TAC increases linearly with respect to time until a pulse is received from the photomultiplier, at which point the TAC stops. The resulting voltage is measured, digitized, and stored in a memory location of a multichannel pulse height analyzer (MCPHA) which corresponds to the magnitude of the TAC voltage. Thus, each memory location or channel in the multichannel analyzer is proportional to time. A decay curve in time can then be inferred from the number of counts per channel, in effect creating a histogram of the decay profile of the luminescent sample.

The time-of-flight single-photon-counting technique possesses greater sensitivity than the phase fluorometer because of its ability to count single photon events. However, two difficulties arise in its application. The first problem is that the content of the MCPHA is actually the probability of time of arrival of a photon at the photomultiplier tube. The TAC circuit will shut off when the first photon is received per lamp flash. If two photons from the sample arrive at the

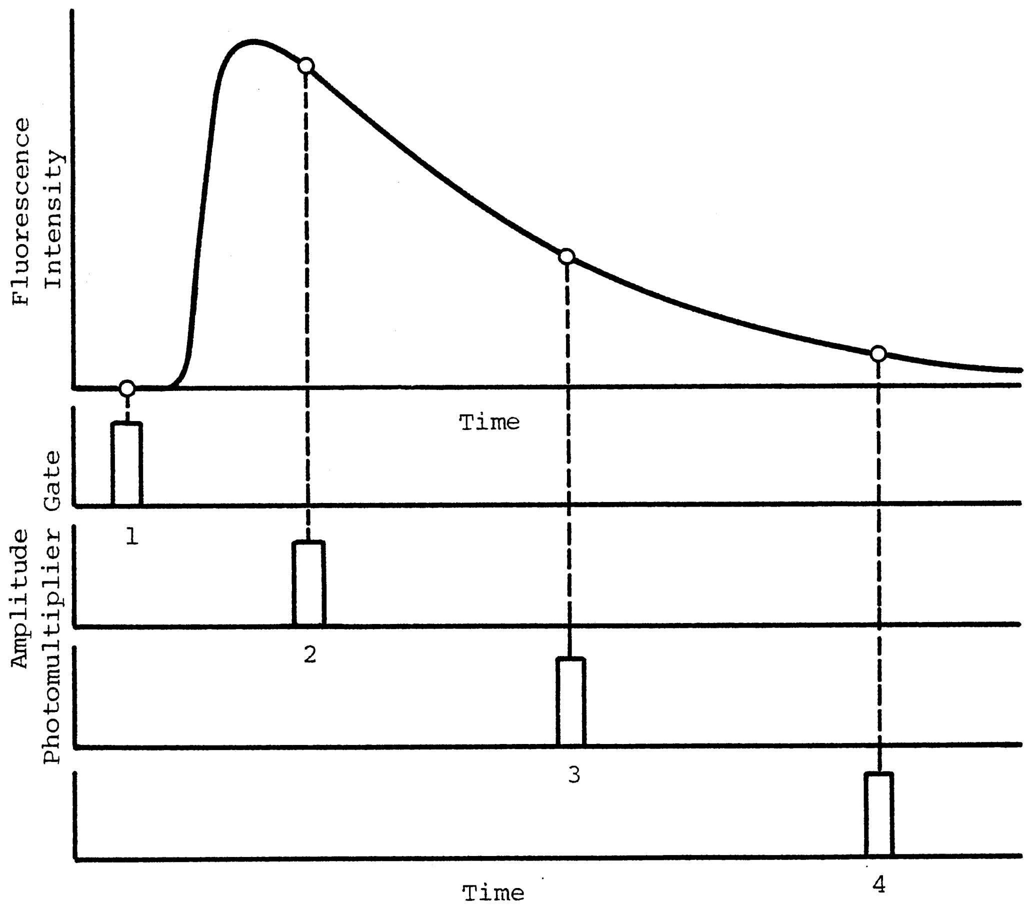

Figure 2. Variable Delay Timing

detector per lamp flash, then only the time of flight of the first photon will be measured. The recorded statistical average of photon pulse arrival will thus be preferentially weighted to give a shorter lifetime than is actually the case. This phenomenon is referred to as "pile-up". To avoid this problem, one must have a count rate of one photon or fewer arriving at the photomultiplier per 100 lamp flashes (1, 20, 21). The second problem is inherent in the nature of basic measurement of decay times by pulsed methods. The problem is that the decay curve is distorted by the finite width of the lamp flash. In a typical free running nitrogen lamp, the full-width-half-maximum of the lamp flash (FWHM) is usually on the order of one or two nanoseconds. If the decay time of the fluorescent species is short relative to the lamp profile, then one observes a decay profile with molecules which are not all excited at the same relative time. That is, the decay profile is convolved with the profile of the lamp.

The mathematical technique used to eliminate this type of distortion, as well as distortion incurred by the detection apparatus, is called deconvolution (1, 7, 20, 22). Deconvolution for single and multiple exponential decays does not necessarily require one to know the functional form of the decay (12). One early mathematical method tried that did not require knowledge of the functional form of the decay was a Fourier transformation. However, the Fourier transformation approach suffers from the fact that at particularly long times instabilities occur due to random fluctuations in real data (13, 14, 15). Fourier transformations are suitable to numerical techniques only when the form of the function to be analyzed approaches zero at both extremities. In single photon counting techniques the decay curve extends to infinity.

If the data is truncated, a discontinuity occurs which does not favor treatment with the Fourier method.

Laplace transformations similarly do not require the functional form of the decay. LaPlace transforms, as with Fourier transformations, suffer from non-discontinuity problems at long lifetimes. These discontinuity problems at long lifetimes can be eliminated by the use of a trailing function to simulate the decay curve at the extreme. This, however, must be considered only an approximation to the curve (17).

The best method for deconvolving lifetimes of single exponential decays measured using pulsed techniques is the phase-plane method proposed by Demas and Adamson (16). If the decay is exponential in character, then it can be converted into a linear form. If the instrumental output is given by the convolution theorem, then

$$I(t) = \int_o^t KG(\lambda)F(t-\lambda)d\lambda$$

where I(t) is the observed luminescent decay signal, G(t) is the fluorescence decay function, and F(t-λ) is the excitation lamp flash function. The linear form is

$$Z(t) = -\tau W(t) + K\tau \tag{14}$$

Z(t) and W(t) were expressed as,

$$Z(t) = \frac{\int_o^t G(\lambda)d\lambda}{\int_o^t F(t-\lambda)d\lambda} \tag{15}$$

and

$$W(t) = \frac{G(\lambda)}{\int_o^t F(t-\lambda)d\lambda} \tag{16}$$

By obtaining the values of G(t) and F(t-λ), a plot of Z(t) versus W(t) can be constructed which will yield a slope of -τ and an intercept of kτ. Demas and Adamson state that the advantages in using this mathematical model are that the phase-plane method can be done with minimal calculation effort and that the linearity of Z(t) versus W(t) is a check on the assumption that the real decay is of single order (10, 20).

Greer, et al. (10) tested the accuracy and precision of the phse-plane method using simulated lifetime curves. The results were compared with the actual values and also to data analyzed by Laplace and Fourier transformations. Table I shows the results found in using lamp and decay functions of varying amplitudes. It was found that for a single exponential fit, the phase plane method was trival because the method was free from cutoff corrections as apply to Laplace and Fourier methods. The phase-plane method values were also found to be relatively insensitive to that portion of the curve that was to be fit. Greer, et al. (10) concluded that even for multiple decay kinetics the method may provide initial parameters for a least-squares decay-curve fit.

The values shown in Table I by Greer, et al. (10) give a low relative percent standard deviation for lifetimes lower than 100 picoseconds. It must be realized that if some type of experimental single photon counting apparatus could be built to measure these picosecond lifetimes, then the errors as calculated by the phase-plane method would be those as listed in Table I. Table I shows that as the lifetime decreases the error associated with the method rises dramatically.

For species with multiexponential decays, a choice of three mathematical techniques is available to correct for the lamp function. The three techniques are the curve fit deconvolution method, the Laplace

TABLE I

DECONVOLUTION BY THE PHASE PLANE METHOD

Run Number	Tau, Picoseconds	10,000 Counts-Lamp & Decay	
	Actual Lifetime	% Rel. Error	% Rel. Standard Error
1	2.5	8.2	28.4
2	5.0	-6.1	14.8
3	10.0	-1.5	8.5
4	25.0	0.0	6.7
5	50.0	-0.6	3.6
6	100.0	-0.3	2.4

transformation method, and the method of moments. The deconvolution method by Hundley, Coburn, Garwin, and Stryer (19) and by Ware, Doemeny, and Nemzek (20) uses a model function which has been normalized and curve fit (25). In this curve fit method, the decay function is defined as an integral equation which is composed of a series of single exponentials. The decay function can be shown to be (20, 22)

$$F(t) = \int_o^t G(t)I(t-t')dt' \quad (17)$$

where

$$G(t) = \sum_{i=1}^{n} a_k e^{-t/\tau} \quad (18)$$

and I is the lamp function, F the decay function, and τ the mean natural radiative lifetime. The values of a_k and τ are guessed and a fit is calculated. Deconvolution by expansion of the exponential form is a type of least squares curve fitting (7). The technique involves using Equation (17) and initially guessing an a_k and a τ. The next value of a_k is chosen such that I (last calculated) can be determined from:

$$\sum_{i=1}^{m} \rho\omega_i (2P_i/2a_k) = 0 \quad \text{for } \ell = 1,2,3, \ldots, n \quad (19)$$

where,

$$\rho = I(\text{observed}) - I(\text{last calculated}) \quad (20)$$

$$W_L = \text{the weighting factor} \quad (21)$$

$$I(\text{last calculated}) = \int_o^t \Sigma a_k e^{-t'/\tau} F(t-t')dt' \quad (22)$$

Equations 17, 18 and 22 give a series of simultaneous equations in which the elements are,

$$D_{k\ell} = \sum_{i=1}^{m} W_i F_{ik} F_{i\ell} \tag{23}$$

$$B_\ell = \sum_{i=1}^{m} W_i F_{i\ell} \; I(\text{observed})(t) \tag{24}$$

$$D_{k\ell} \, a_\ell = B_\ell \tag{25}$$

and,

$$F_{ij} = \int_o^t e^{-t/\tau_j} \, I(t_i - t')\,dt \tag{26}$$

The sum of the square of the residuals from the errors in the curve fit is calculated and lifetime values and amplitudes are produced. The lifetime value is then varied until the sum of the square of the residuals is at a minimum. This deconvolution method was tested by writing a deconvolution routine on an IBM 370/168. This method assumes that almost any decay law can be represented by a sum or difference of exponentials if enough exponentials are used. From a fit of the exponentials it can be concluded that no physical or mechanistic significance should be attached to the amplitudes.

The problem with this technique is the use of fixed exponential coefficients (7, 21). It was concluded that if too few exponentials were used, an inaccurate fitting of the real function occurs. Also, if too many exponentials were used then random fluctuations in the data leads to wild oscillations in the fitting at longer lifetimes.

Recently, Birch and Imhoff (23) used the curve fit method to analyze the instrument pulse profile from the photomultiplier. In measuring real samples of DPH in propylene glycol between -50 and +70 degrees centigrade it was found that the fit gave a single exponential fit with a resolution of 100 psec. The least squares correlation coefficient was 0.997.

A second approach to deconvolution for multiple-decay kinetics transforms the model function into a linear equation using Laplace tranforms (10, 18).

A Laplace transformation impulse function of a multiple decay law can be given as,

$$G(t) = \sum_{i=1}^{n} a_i e^{-t/\tau_i} \quad (27)$$

For single exponentials if one defines the Laplace transformation of a function F(k) as,

$$F(k) = L[I(t)] = \int_{o}^{\infty} \sum_{i=1}^{n} m(t) e^{it} dt \quad (28)$$

then the transform of the function is just,

$$L[M(t) e^{-t/\tau}] = \sum_{i=1}^{n} \frac{A}{k + 1/\tau} \quad (29)$$

From Laplace convolution theory it can be shown that the decay function is equal to the transform of the lamp times the impulse function.

$$F(k) = L[\int_{o}^{t} G(t-\tau) \cdot I(t') dt'] = L[E(t)] \cdot L[I(t)] = E(k) \cdot I(k) \quad (30)$$

By combining Equation (29) and Equation (30) it can be seen that,

$$\frac{F(k)}{E(k)} = I(k) = L[I(t)] = \sum_{i=1}^{n} \frac{A}{k + 1/\tau} \quad (31)$$

In practice the lifetime values obtained by Laplace transformations are solved by calculating the decay and lamp curves for the 2n different values of k used. The set of 2n equations are solved to yield n decay constants and n amplitudes. The solution is then obtained by a procedure of iterative reconvolution. After the difference in coefficient values reaches a small difference then the reconvolution can be stopped.

The disadvantage in using Laplace transformations for long decay times is that a correction cutoff must be made. This is because the Laplace transformation is defined from time zero to time infinity. For long decay lifetimes the decay does not vanish to zero at long times and if a correction is not made the calculated lifetime value will be smaller than the true value (18).

A computer program was written to estimate errors associated with typical data using the Laplace method if no cut off correction was to be added. The errors ranged from one to ten percent for lifetimes between one and ten nanoseconds.

The third deconvolution method commonly used involves the method of moments (25, 26). If the moments of the lamp and decay are defined as,

$$\mu_k = \int_o^\infty t^k F(t-\lambda)d\lambda \quad (32)$$

$$m_k = \int_o^\infty t^k G(\lambda)d\lambda \quad (33)$$

then the moments define a set of linear equations.

$$\begin{aligned} \mu_1 &= G_1 m_o \\ \mu_2 &= G_1 m_1 + G_2 m_o \\ &\vdots \\ \mu_n &= G_1 m_n/n! + G_2 m_{(n-1)}/(n-1)! + \ldots + G_n m_o \end{aligned} \quad (34)$$

These equations are used to solve for an N component system where,

$$I(t) = \sum_{n=1}^{m} a_n \tau \tag{35}$$

The lifetime is calculated from an iterative process. The disadvantage with the method of moments is that there is little warning of the form of the decay kinetics. Also, there are problems associated with taking experimental data with large enough signal-to-noise ratios that good convergence is assured (27).

Because of these mathematical deconvolution problems an alternate method called phase fluorometry has recently gained more popularity. These phase fluorometry techniques have developed independently of the pulse techniques. The first successful phase fluorometer built was that built by Gaviola in 1926 (28-30). The experimental apparatus is shown in Figure 3 uses light modulated by a Kerr cell (32).

In a phase fluorometer a sample is excited by modulated light. After absorption of the light by the sample the molecules relax into the ground state. If the modulated emission is monitored, the phase of the emitted light will be shifted relative to the incident light. The phase shift associated with the time the molecules take to relax back into their ground state is a function of the lifetime of the excited state.

This relation between the luminescent phase shift and the fluorescent lifetime for a simple single exponential decay was first derived by Gaviola (28). More complicated kinetic mechanisms were determined later by Birks, Dyson and Munro (17). In 1970, Spencer (31) derived a relationship for double exponential decays that stresses the importance of a simultaneous measurement of both the modulation depth and phase shift of the emitted light. The fluorescence lifetime can be obtained from

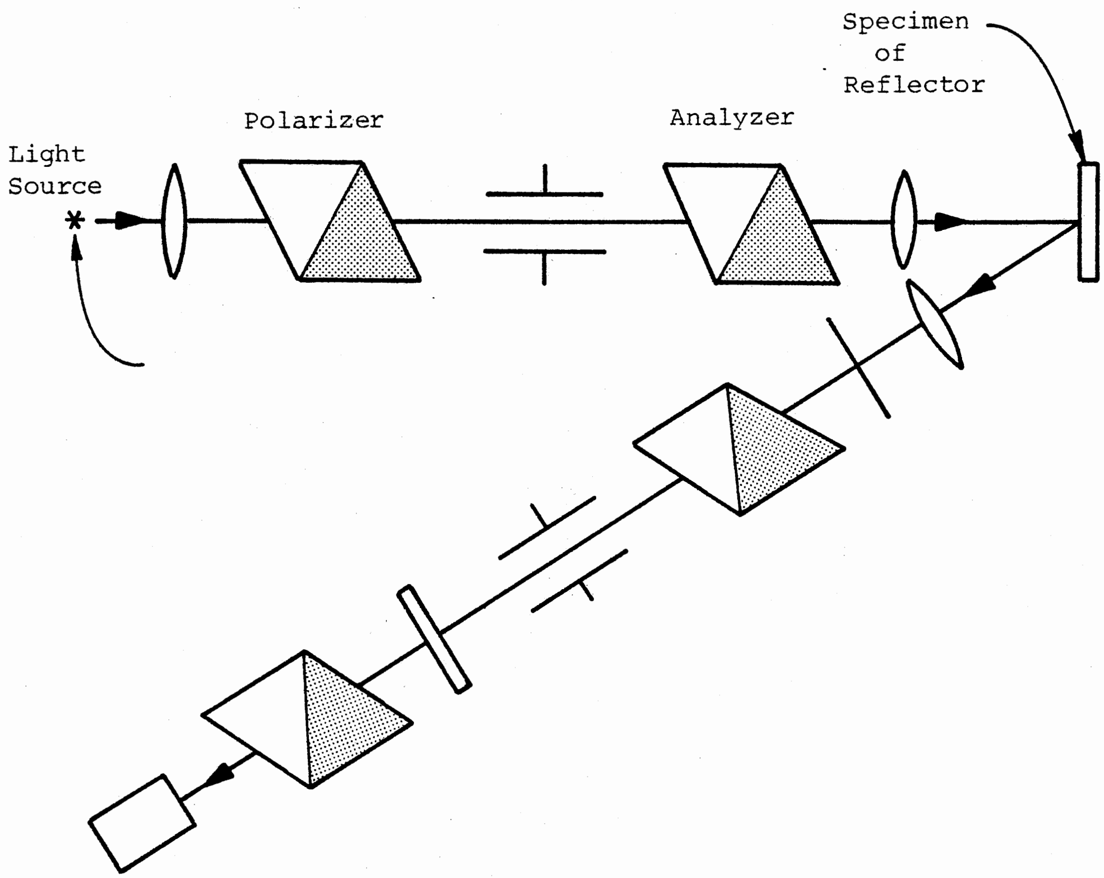

Figure 3. The Phase Fluorometer of Gaviola

either measuring the phase shift or the relative modulation depth for a single exponential. Both measurements are required to check that the decay is single exponential. If the decay is truly single exponential in character then the lifetime values from each measurement will be the same. For simulated data it was shown (39) that a variation in the phase lag as shown in Figure 4 indicates that only in the case of a single exponential decay will the lifetime measured by phase shift and by modulation depth will be the same values. Figure 4 shows two examples of phase shift values at modulation frequencies of 14.2 MHz and 28.4 MHz (31).

Earlier it was shown that if species A is excited by a sinusoidal modulation function f(t) then the differential equation becomes:

$$\frac{d[A^*]}{dt} = -k[A^*] + f(t)$$

If the exciting function is sinusoidal then Equation (36) can be described as,

$$\frac{d[A^*]}{dt} + k[A^*] = A + B\sin\omega t \qquad (37)$$

Solving the differential equation yields,

$$[A^*] = x + y\sin\omega t + Z\cos\omega t \qquad (38)$$

where,

$$\frac{d[A^*]}{dt} = y\omega\cos\omega t - Z\omega\sin\omega t \qquad (39)$$

If one now substitutes Equation (38) and (39) into Equation (37) one

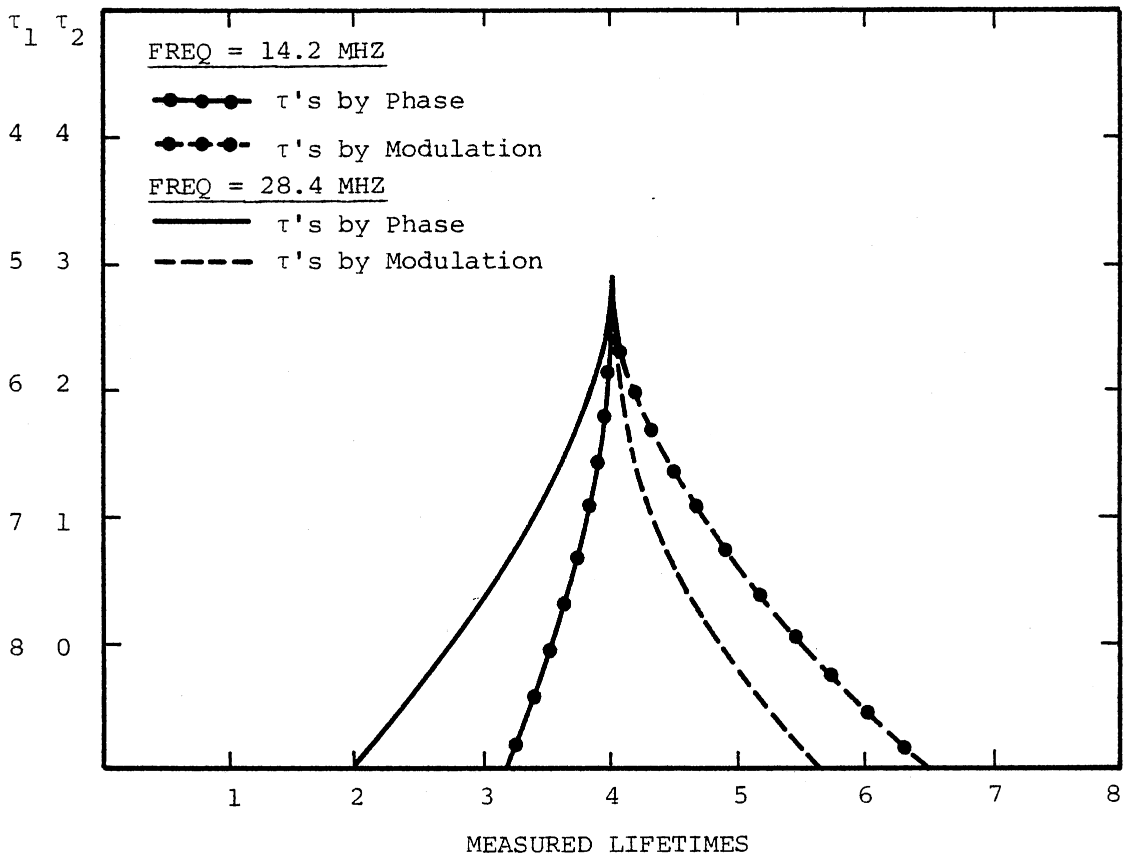

Figure 4. Phase Shift Versus Modulation Measurements

obtains,

$$y\omega\cos\omega t - Z\omega\sin\omega t + X\lambda + y\lambda\sin\omega t + Z\lambda\cos\omega t = A+B\sin\omega t \qquad (40)$$

Equation (40) can be separated into A.C. and D.C. components. Solving for the D.C. components one obtains Equation (41).

$$X\lambda = A \qquad (41)$$

or,

$$X = A/\lambda$$

When one separates the A.C. components to solve for y and z, one obtains,

$$y\omega\cos\omega t - Z\omega\sin\omega t + y\lambda\sin\omega t + Z\lambda\cos\omega t = B\sin\omega t$$

$$(y\omega + Z\lambda)\cos\omega t + (y\lambda - Z\omega)\sin\omega t = B\sin\omega t \qquad (42)$$

If we set the conditions such that wt = $\pi/2$ radians then,

$$(y\omega+Z\lambda)(1) = 0$$

$$y = -Z\lambda/\omega \qquad (43)$$

Substituting Equation (43) into Equation (42) and solving for z we obtain,

$$(-\frac{Z\lambda\omega}{\omega} + Z\lambda)\cos\omega t + (-\frac{Z\lambda^2}{\omega} - Z\omega)\sin\omega t = B\sin\omega t$$

$$-Z(\frac{\lambda^2}{\omega} + \omega)\sin\omega t = B\sin\omega t$$

$$-Z(\frac{\lambda^2}{\omega} + \omega) = B \qquad (44)$$

If one now divides both sides by (λ)**2 the result is,

$$z = \frac{-B\omega}{\lambda^2(1 + \omega^2/\lambda^2)} \tag{45}$$

Resolving for y yields,

$$y = \frac{B\omega}{\lambda^2(1 + \omega^2/\lambda^2)} \cdot \frac{-\lambda}{\omega} \tag{46}$$

Therefore, the results of x, y, and z are,

$$x = \frac{A}{\lambda} \; ; \; y = \frac{B}{\lambda(1 + \omega^2/\lambda^2)} \; ; \; z = \frac{-B\omega}{\lambda^2(1 + \omega^2/\lambda^2)} \tag{47}$$

If one now substitutes the results of Equation (47) into (38) we obtain,

$$[A^*] = \frac{A}{\lambda} + \frac{B}{\lambda}\left(\frac{1}{1 + \omega^2/\lambda^2}\right) \sin\omega t - \frac{B\omega}{\lambda^2}\left(\frac{1}{1 + \omega^2/\lambda^2}\right) \cos\omega t \tag{48}$$

Upon making the following two substitutions,

$$\cos\phi = \frac{1}{\sqrt{1 + \omega^2/\lambda^2}} \; ; \; \sin\phi = \frac{\omega}{\lambda\sqrt{1 + \omega^2/\lambda^2}} \tag{49}$$

Equation (37) becomes,

$$[A^*] = \frac{A}{\lambda} + \frac{B}{\lambda}\cos\phi[\cos\phi \sin\omega t - \sin\phi \cos\omega t]$$

$$[A^*] = \frac{A}{\lambda} + \frac{B}{\lambda}\cos\phi[\sin(\omega t + \phi)] \tag{50}$$

The modulated emission, E, is just the rate constant times the concentration of the excited species and the variable b is the modulation coefficient.

$$E = \lambda[F^*] = a+b\ \sin(\omega t+\phi) \tag{51}$$

$$a = A/\lambda \tag{52}$$

$$b = (B/\lambda)\cos\phi \tag{53}$$

From Equation (37) it can be seen that the tangent of the phase angle is related to the lifetime.

$$\tan\phi = \frac{\sin\phi}{\cos\phi} = \frac{\omega}{\lambda} = \omega\tau \tag{54}$$

The modulation of the wave from Equation (53) is defined as,

$$M = \cos\phi \tag{55}$$

The modulation can be related to the lifetime by making several simple substitutions.

$$\tan\phi = \omega\tau$$

$$\tan\phi = 2\pi\nu\tau \tag{56}$$

Upon squaring Equation (56) and substituting the definition of the tangent function, one can obtain,

$$(2\pi\nu\tau)^2 = \tan^2\phi$$

$$(2\pi\nu\tau)^2 = \sin^2\phi/\cos^2\phi$$

$$(2\pi\nu\tau)^2 = (1 - \cos^2\phi)/\cos^2\phi \tag{57}$$

Substituting Equation (55) into (57) yields,

$$(2\pi\nu\tau)^2 = \frac{1}{M^2} - 1$$

$$\tau = \frac{1}{2\pi\nu}\sqrt{\frac{1}{M^2} - 1} \qquad (58)$$

Thus, it is possible to determine the lifetime from the phase shift or modulation parameters. From Figure 5 the phase shift is just the difference in phase between the sinusoidal wave before reaching the sample compared with the phase of the sinusoidal wave radiating from the sample. The measurement of the relative modulation depth is just the maximum intensity of the modulation depth of fluorescence divided by the maximum intensity of the modulation depth plus the component of the signal, where

$$\text{Modulation Depth} = \frac{(F_{max} - F_{min})/(F_{max} + F_{min})}{(S_{max} - S_{min})/(S_{max} + S_{min})} .$$

The measurement of the relative modulation depth and phase angle was first achieved by Gaviola in 1926 (29). From the earliest development of phase fluorometry, modulated sources were designed using novel techniques. The instrument designed by Gavioli used a polarizer to pass light through a Kerr cell (29, 30). A second polarizer and Kerr cell were used for the emission. Thus, the signal received through the second polarizer is phase sensitive to the first if the Kerr cell is driven by a voltage source. An analyzer is typically put in the optical path to adjust the delay distance. When the analyzer is shifted and the signal is a maximum then the decay is obtained. The sample is replaced by a mirror in order to calibrate the analyzer due to the fact that the optical path

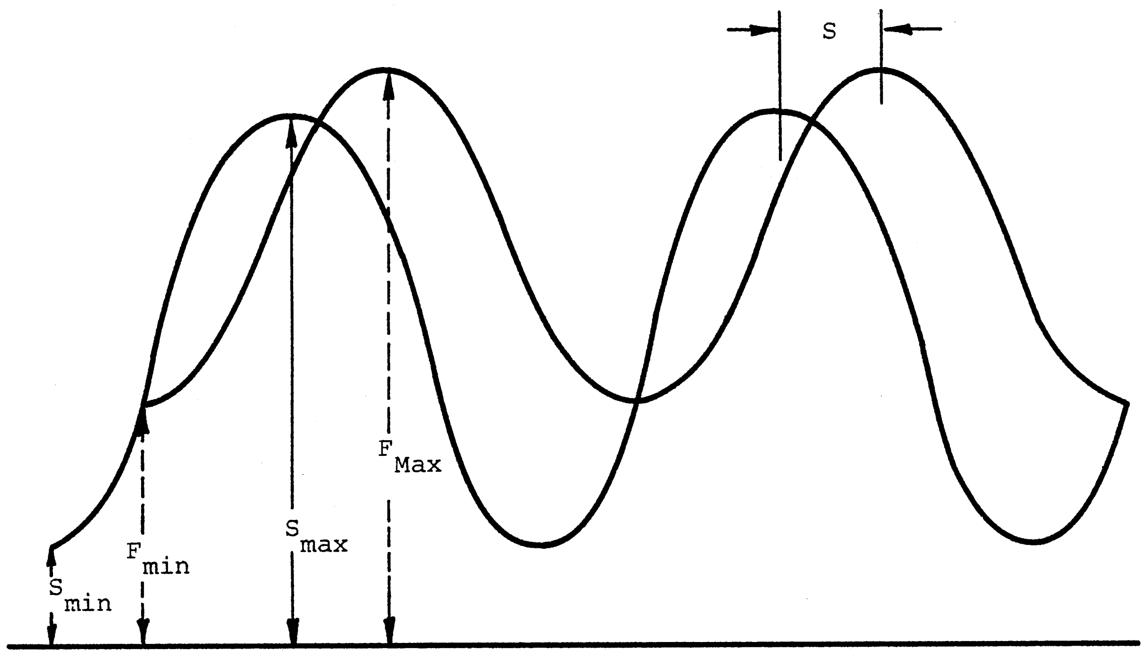

Figure 5. Phase and Modulation Depth Measurement

length is shifted a known distance.

The Kerr cell (32) is a tank filled with liquid across which a potential field is applied. A modulating voltage is applied to the two electrodes and a solvent, usually nitrobenzene, rotates the plane of polarization of the beam of light. When two polarizers are on either side of the tank and are perpendicular to each other, then a pulsed beam is created. Kerr cells have found application as high speed shutters and are effective replacements for rotating wheels. In a cell the rotation of the plane of polarized light is,

$$\Delta n = \lambda K E^2$$

where E is the potential voltage in statvolts (1 statvolt = 300 volts) and K is known as the Kerr constant. Table II shows several Kerr constants for some selected liquids. Benzene which has the lowest dipole moment has the lowest constant. Nitrobenzene which has the highest dipole moment has the highest constant. Two problems, however, are evident with the Kerr cell. The first is that because the electrodes are usually separated by one centimeter for optical reasons a rather large voltage is required for the cell to respond. The second problem with Kerr cells is that nitrobenzene is toxic and explosive (32).

Later, it was observed that the same polarizing effect could be achieved using crystals which lack a center of symmetry (7). The advantage here is that the cells did not contain nitrobenzene and the applied field is nearly an order of magnitude less. This electro-optic effect is known as the Pockels effect. The crystal used most widely today is potassium dideuterium phosphate (32). The response of this crystal is fast and the field voltage low. The first phase fluorometer

TABLE II

KERR CONSTANTS OF SOME SELECTED LIQUIDS

Molecule	Kerr Constant (10**-7 cm/statvolt**2)
Benzene	0.6
Carbon Disulfide	3.2
Water	4.7
Nitrotoluene	120
Nitrobenzene	220

using a Pockels cell was built by Lumry (33). The instrument was driven at a frequency of less than 10 MHz because of problems with RF leakage and cooling. The problem with Pockels cells in general is that the source must be between one and 3,000 volts RMS. A sensitive RF detector, such as a high tension phototube, used for detecting the phase shift of the light emitted from the sample will pick up the RF oscillation of the modulating source in addition to the photon generated signal (35). Thus, although Pockels cells offer some advantages over Kerr cells, the applied AC voltage is still high. To remove this requirement of a high AC driving voltage. Spencer and Weber (31) used a Debye-Sears ultrasonic tank. The smaller AC voltage required (ca. 5 volts RMS) can be an advantage in the elimination of RF pickup by the detector. The frequency doubling of the modulated light diffracted by this tank is an added advantage because the tank modulator is operated at half the frequency of the RF detector which greatly eliminates coherent signal pick-up.

The ultrasonic liquid tank uses an effect initially characterized by Debye and Sears in 1932 (34). When light passes through the liquid tank the light is modulated by the ultrasonic standing wave generated within the liquid by application of an AC field (32). Tank stability has been achieved by using a mixture of 19 percent absolute alcohol mixed with 81 percent distilled water (7). Spencer and Weber (39) state that this solution has a small temperature coefficient when used for modulation frequencies of up to 28 MHz.

When the liquid is sonicated near 10**7 MHz and an acoustic wave is set up between the transducer crystal and the reflecting plate, then the liquid tank will behave as an optical diffraction grating in which the zero and first order diffraction patterns can be observed.

Maercks (36) was the first to successfully use the standing wave diffraction pattern generated between a quartz crystal and a metal reflecting plate. The choice of liquid in the cell was usually water because it does not absorb in the ultraviolet spectrum. The measurement made was a comparison of the galvanometer reading as a function of path length for the fluorescent sample and the reflecting plate. The typical resolution for instruments such as Maercks was 200 psec.

Although many methods have since been devised to determine the phase angle shift between the exciting light and the fluorescent sample, most techniques fall into one of two basic categories. The first is a phase shift comparison before amplification and the second is amplification of signals before phase shift comparison. Maerck's apparatus falls into the first category and the apparatus of Lumry, et al. (33) falls into the second.

Lumry, et al. (33) used a beam splitter and divided the modulated light beam into a reference and a sample path. Each beam path was detected by an independent photomultiplier in which the signal was measured by a phase sensitive null detector circuit. The principle of the circuit is that when the sum of the output of the reference and sample signal is 180 degrees out of phase, then the output of the phase null detector would be zero. A calibrated delay line is used to zero out the signal. The difference in the time-of-flight between the calibrated delay line and the sample is the lifetime of the excited state species.

In the instrument used by Lumry, et al. (33) serious problems are incurred in the determination of the null point due to noise levels typically encountered in photomultiplier tubes. These noise levels are associated with counting noise and with amplifier and power supply drift. Noise is also picked up through the RF pickup of the high voltage AC signal used in the pockels cell. Bailey and Rollefson (37) overcame this difficulty by heterodyning the signal. They used a second crystal controlled oscillator and adjusted the frequency from their signal by approximately 100 Hz. The signal and reference were then mixed in a nonlinear amplifier producing a signal with the required phase and modulation information at the difference frequency. The difference frequency was then amplified by an audio amplifier and by a narrow band pass twin-T amplifier. Signals were compared and nulled to obtain the phase shift. An advantage of this arrangement was that the direction band pass can be reduced to 5 to 10 Hz, thereby, increasing signal-to-noise ratio. Although Bailey and Rollefson did increase their signal-to-noise ratio, they unfortunately discarded the D.C. component of the signal as well. The D.C. component contains information about the depth of modulation. Birks and Little (38) were able to obtain the D.C. signal by transferring the information of the high frequency signal to a low frequency, in a cross-correlation procedure. To obtain the D.C. information they applied an A.C. standing wave voltage of fixed frequency to one of the dynodes of the photomultiplier tube. The signal obtained from the photomultiplier is then a cross product of the actual signal and the applied A.C. signal with a fixed phase (ϕ). R(t) is the reference frequency applied and F(t) is the fixed frequency from the phototube. δ is the relative phase angle between the two applied frequencies.

$$R(t)\cdot F(t) = aA + bB \frac{\cos\delta}{2} \cos(\phi - \delta)$$

Thus, Birks and Little determined the value of the phase shift by altering the phase, δ, of the A.C. signal until the maximum photocurrent was obtained.

Spencer and Weber (39) used the advantage of cross-correlation detection by varying the phase angle of the applied A.C. signal linearly with time.

$$R(t)\cdot F(t) = aA + bB\,\frac{\cos\delta}{2}\cos(2\pi\Delta ft-\delta)$$

where,

$$\phi = 2\pi\Delta ft$$

The advantage of linearly varying the A.C. phase is that one obtains a low frequency response for the phase difference while still retaining the relative modulation information. The relative modulation is retained due to the mixing of the high frequencies with the applied frequency. The frequency mixing occurs in the photomultiplier tube. After mixing occurs the low frequency measured component contains the modulation depth information.

The cross-correlation method gives basically two advantages. First, frequencies above the difference can be filtered out by low pass filters before audio amplifiers are used to increase the signal amplitude. Secondly, the signal information is transformed to frequency differences of less than 100 Hz. Numerical time counting techniques can then be used to increase the accuracy of the data analysis (39).

A commercial phase fluorometer using cross-correlation techniques is available today. The SLM instruments SLM-400 uses a 450-W xenon lamp modulated by a Debye-Sears modulator tank at 6, 8, and 30 MHz. The monochromator dispersion is 2 nm/mm with stray light at 0.0005%/nm outside the bandpass. The UV transmitting optics are rated with a focal speed of one. The data is collected with a microcomputer and the output is printed on a plotter. The lifetime resolution of the instrument is rated at 10 psec (40).

In an attempt to improve the signal-to-noise ratio for low-light-intensity samples using phase fluorometry Schlag et al. (41) developed a multi-point correlation technique. The advantage of this procedure is that excellent time resolution can be achieved without the inherent disadvantages of pulse techniques discussed previously. Mathematical deconvolution procedures are not required. In the instrument of Schlag et al. (41) the light beam was modulated by an ultrasonic cell driven by a quartz transducer. The reference and sample photomultiplier output responses were shaped to standard TTL logic pulses and were accumulated in a counter. The number of accumulated pulses is correlated versus the reference signal through the use of fast ECL switching gates. The reference signal is obtained from a light bypass. The output of the reference photomultiplier is filtered and delayed by a time phase-locked loop circuit from which the signal is squared. The squared signal provides the reference input to the ECL gating circuit. The difficulty with the method used by Schlag is that the subunits are not commercially available.

More recently, Gugger and Calzaferri (42,43) have used the cross-correlation concept to achieve a precision of 10-20 picoseconds for a nanosecond decay. A beam was split through the use of a beam splitter. One of the optical paths of the split beam was longer, causing a delay.

The beams were rejoined by the use of fibre optics and were guided to a nonlinear multiplier, the photocathode of a photomultiplier. The anode current was Fourier transformed by a spectrum analyzer and a microcomputer. While the precision is good, the problem with this method is the high cost of the instrumental apparatus.

A modification proposed by Chandler (44) uses a time-to-amplitude converter to make the instrument more versatile. The advantage here is that the histogram of pulses counted by the method of Schlag are stored in the multichannel pulse height analyzer. The information can be read out and correlations performed automatically on line. Chandler states that this technique gives "satisfactory performance with nanosecond resolution" (44).

These recent modifications on phase fluorometers have enabled picosecond resolution in several areas of photophysics. This resolution proves useful in the study of solvent cage relaxation and proton transfer of acids, for example (48). Hauser and Heidt (46) studied the kinetic aspects of 2-napthol/naptholate proton change with fluorescence. The time resolving power was stated to be ten picoseconds at a modulating frequency of 72 MHz. Heidt (47) studied the lifetime of thionine in methanol at low concentrations (approx. 10^{-8} M) to determine the quantum yield using a 514 nm laser line. Shapiro and Winn (49) studied the solvent cage effects of acridine in solution using a variety of solvents. They obtained a resolution of 50 psec. In this paper they rule out a hydrogen bonding mechanism for deactivation of a higher triplet state of acridine as proposed by Huber et al. (50). Their contention was that because no effect was observed on the lifetime at concentrations of 10^{-6} molar then the role of dimers as proposed by Huber et al. may be ruled out. In fact, lifetime measurements with precision of less than or equal to one picosecond are required to substantiate this argument.

Other studies include that of Link (51) in which he studied the radiative lifetimes of the first excited states of Na, K, Rb, and Cs with a phase fluorometer (52). In 1969, Schlag and Weyssenhoff (53) studied the relaxation of vibronic states of β-napthylamine as the energy was varied from 390 to 344 nm. They determined that the excited state lifetime was on the order of 10^{-8} seconds.

Although much progress has been made in the construction of fluorometers since the days of Gaviola, the technique of successfully measuring lifetimes with picosecond resolution is still difficult. Phase and pulse techniques each offer independent advantages. The phase method gives better resolution while the pulse method generally offers easier detection of multiple decay kinetics (54). It therefore remains to develop a high resolution fluorescent spectrometer that will enable one to easily observe single and multiple decay phenomenon with time resolution and precision better than one picosecond, if possible. If single photon counting sensitivity could be included, that would give the best hybridization of all previous generation instruments.

CHAPTER II

THE SINGLE PHOTON COUNTING METHOD

Time-dependent molecular vibrations of molecules have been studied for many years. Lifetimes have been measured to characterize lipid phase membranes of vesicle systems. The fluorescent lifetime of an organic dye probe can be used to describe the rotational motions of the probe in a lipid environment. By monitoring the depolarization of light, the rotation of a molecule in its local environment can be seen. One can thus determine the rotation of the emitting dipole in the lipid membrane environment. With this information one can also predict kinetic relaxation mechanisms.

The basic apparatus used in this work for lifetime measurements of lipid-dye membrane systems is shown in Figure 6. The lamp used was a nitrogen spark lamp. The potential across the lamp was 900 V and the nitrogen pressure was 10 psi. The lamp firing rate was on the order of 10 KHz. A lamp pickoff signal was sent to an Ortec Model 473A constant fraction discriminator. The output of the constant fraction discriminator provided the start pulse for a Le Croy QVT Multichannel Model 3001 Analyzer, time to amplitude converter combination. The lamp wavelength was tuned by a PTR minichrome 1 monochromator. The sample chamber consisted of a mount for a quartz cuvette which was centered in the chamber. An option also existed for inserting a temperature controlled water jacket around the sample. This was to monitor temperature effects on the

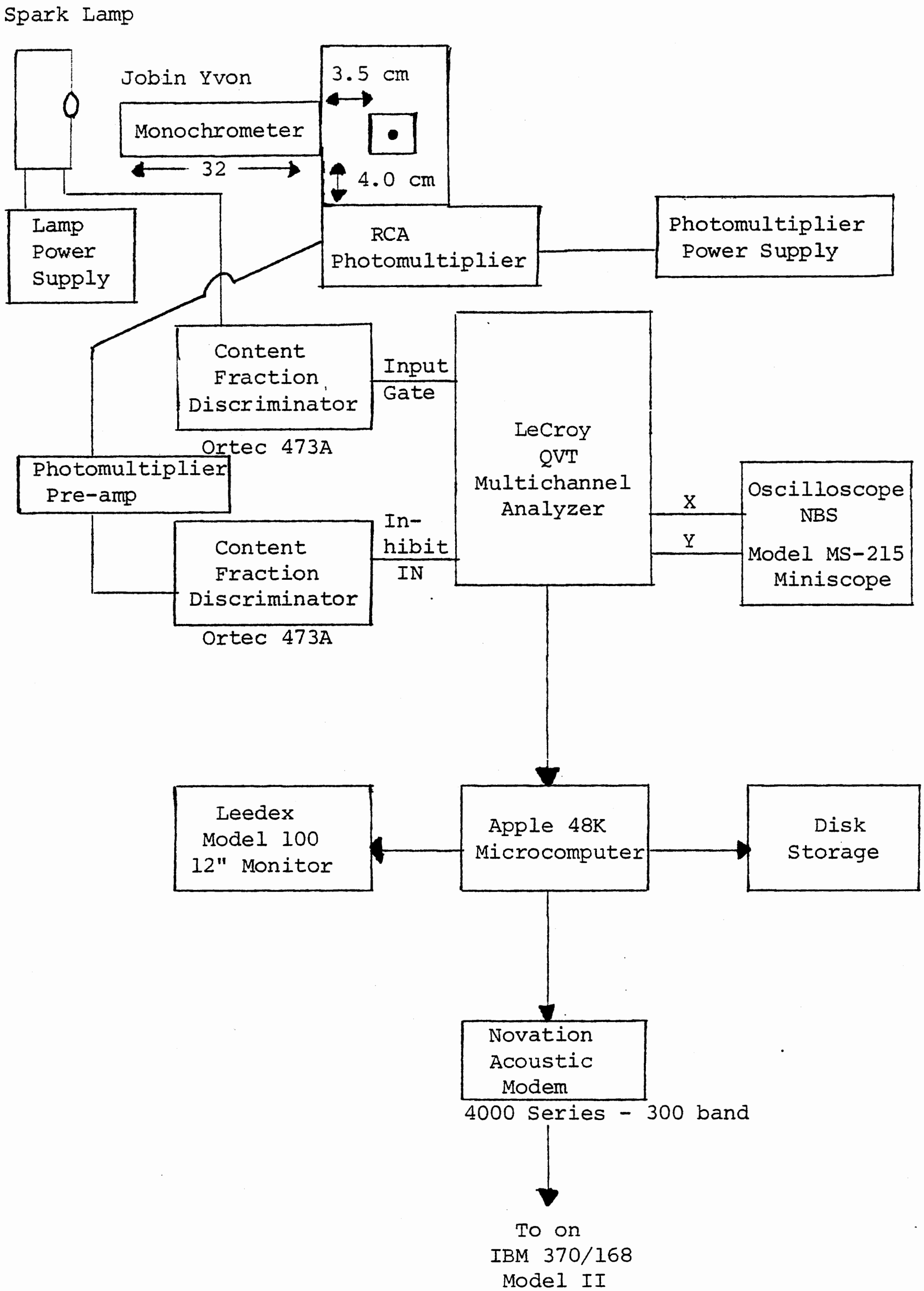

Figure 6. Single Photon Counting Apparatus

time dependent depolarization ratio and observe the effects by the probe upon the lipid membrane.

The detector used was an RCA 4084 9 stage photomultiplier tube. The photomultiplier was powered by a Model 226 Pacific Photometric instrument photomultiplier power supply at 900 volts. The output from the RCA photomultiplier tube was fed into a Pacific Photomultiplier Model 9031 preamplifier. The output of this preamplifier was then fed into an Ortec 473A constant fraction discriminator. This constant fraction discriminator output was used as the stop pulse for the Lecroy QVT Multichannel Analyzer.

An NLS Model MS-215 miniscope was connected to the multichannel analyzer to provide a display of the data. The data was digitized and stored on an Apple II 48K computer. An interface board was built that would interrupt the clock in the multichannel analyzer. A pulse strobe was then sent from the Apple microcomputer to the multichannel pulse-height analyzer. When the multichannel pulse-height analyzer was pulsed, a digital signal could be obtained from the buffer in the multichannel analyzer on the output pins of a 40-pin connector.

The Apple microcomputer also was connected to a Novation 4000 Series Acoustic Modem through a 300-band serial interface board. The modem was used to dial up an IBM 370 Model 168 mainframe computer. The Apple was used for data storage and manipulation of the data, while the IBM 370 was used for data processing. Single and double exponential curve fitting deconvolution programs were written for use on the IBM mainframe. A single exponential phase-plane deconvolution method was available on the IBM 370 as well as the Apple microcomputer.

In order to compare lifetime values obtained by the instrument with comparable literature values, several dyes were run in various solvents. A 0.45 optical density solution of Rhodamine 6G was prepared. The solvent used was reagent grade ethanol. The exciting wavelength was 530 nm with the emission wavelength set at 560 nm. The number of counts in the maximum channel of the multichannel analyzer was 1000 counts. The data was analyzed by the phase-plane single exponential fitting program which gave a lifetime for the Rhodamine 6G solution of 3.81 ± 0.1 nsec. The generally accepted lifetime of a 0.04 optical density solution of Rhodamine 6G in ethanol is 3.9 nsec (56).

Another standard solution on which data was taken was that of Anthracene in cyclohexane. A concentrated stock solution of 0.3 g/l was made in cyclohexane and the solution was filtered through a millipore filter. The solution was then diluted until a 0.3 optical density solution was obtained. The absorption maximum of anthracene is 355 nm while the emission maximum is at 400 nm. The lamp curve was obtained at 300 nm while the decay curve was obtained at 400 nm using a filter in place of the excitation monochrometer. The filter used was an Ortec 5127 filter with an optical cut-off at 350 nm. The criteria of choice for selecting the emission filter was the cut-off frequency between the exciting wavelength and the emission wavelength. The filter was used to reduce the effect of scattered light. The anthracene lifetime data was run to 4 K counts in the maximum channel of the multichannel analyzer. The lifetime analyzed by the phase-plane method was 4.63 ± 0.1 nsec. The generally accepted lifetime value given by Berlman (56) is 4.9 nanoseconds.

The two dyes studied, rhodamine 6G and anathracene, showed good agreement with the literature values. The results were especially encouraging because the maximum number of counts in any channel of the multichannel pulse height analyzer was less than 10 K for any experiment. The precision with this, as with most single photon counting machines was that the resolution was several hundred picoseconds. Upon completion of the lifetime measurements of rhodamine 6G and anthracene the lifetime of a probe in a lipid vesicle environment was measured.

1,6 DPH Fluorescent Probe in Lecithin Vesicles

The experiment consisted of observing the fluorescence of 1,6-diphenylhexatriene in two different chemical environments. The object of the study was to determine a saturation point of the specially prepared lipid vesicles with the fluorescent probe. When this saturation value is reached, the resulting fluorescent lifetime should approach a double exponential character. The effect is due to the fact that above saturation of the vesicles the DPH is in two different environments. Below the saturation concentration all of the DPH should insert itself into the membrane and one should obtain only a mono-exponential fluorescent decay.

The lipid lecthin vesicles were prepared by injecting 250 μL of the lipid into a 10-mL solution of phosphate buffer. During the injection the phosphate buffer solution is kept at a temperature well above the phase transition of the lipid. The injection velocity was approximately 0.025 mL/min with the stirring velocity at 0.33 rev/sec (55).

Several concentrations of dye to lipid vesicle concentrations were prepared and the fluorescent lifetimes obtained. From a steady state absorption versus concentration plot at 360 nm, it can be seen that the apparent saturation concentration of the dye into the lipid membrane is approximately 4 μL of dye per 10 mL of lipid phosphate buffer solution.

Thus, if one could monitor the lifetimes of the components above the saturation concentration of DPH, then one should observe the DPH in the two different environments. The DPH should fluoresce in the membrane as well as in the buffer solution. Because the molecule in the membrane is constrained to a certain extent the lifetime would be somewhat longer than the lifetime of the molecule in the buffer (55).

The absorbance values shown in Table III may indicate that the saturation point of probe to membrane ratio has been reached. The saturation concentration of DPH in the lipid membrane is nearly 4 μL. Thus, the absorbance values indicate that the dye should be in two environments while the lifetime measurements indicate that the fluorescent decay is nearly mono-exponential.

The results shown in Tables III and IV calculated by the phase-plane deconvolution technique shows that even at a 7.0 μL concentration, the lifetime is single exponential. This result is contrary to the result from the steady-state fluorescence intensity measurement.

The explanation for this puzzling result lies in the fact that a low total concentration of the dye originally present in the buffer was absorbed by the vesicles. The fluorescent component due to the DPH was very small compared to the buffer component. Due to errors in deconvolving the data caused by a high background component the second exponential component, if present, was never observed.

A major problem with the time-of-flight single photon counting technique is in the interpretation of the data to fit a double exponential curve, one typically deconvolves the data using a curve fitting technique. This involves fitting a sample curve with the coefficients until a good fit is obtained with a minimum of errors as explained

TABLE III

ABSORBANCE OF DPH AT 360 NM

Molar Concentration	Absorbance at 360 nm
1.2 x 10(-6) M	0.106
1.2	0.108
1.75	0.135
2.33	0.195
2.33	0.203
3.5	0.305
4.6	0.340
5.67	0.345
7.0	0.3675

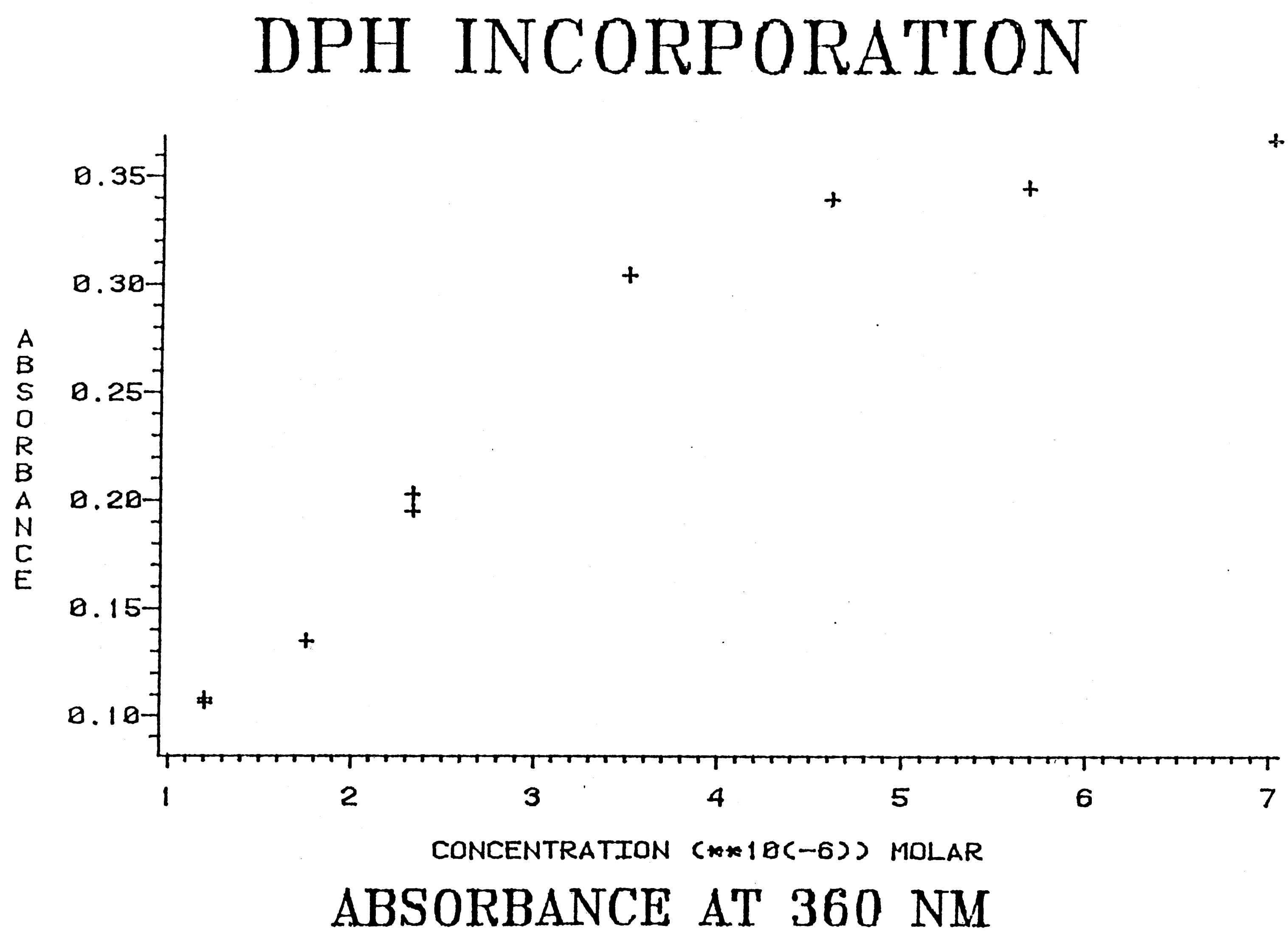

Figure 7. DPH Incorporation of Lipid Vesicles

TABLE IV

DPH LIFETIMES

Conc (μL)	Single Exponential Res.		Double Exponential Res.		
1.2 (**10(-7))	8.11	0.932	-0.139 0.333	11.66 8.94	34.2
2.33	8.09	0.759	-0.480 0.372	3.88 9.66	3.13
2.33	7.80	2.41	-0.804 0.158	11.43 11.43	2.62
3.5	7.38	1.22	-0.18 0.412	3.95 9.72	6.65
4.6	7.06	0.86	-0.103 1.07	6.32 7.70	3.06
5.67	6.31	2.74	0.311 0.117	9.13 9.16	4.17
7.0	7.00	1.50	0.243 0.456	8.93 8.91	10.7
20.0	6.48	1.98	non-exponential		
30.0	7.55	6.36	non-exponential		
40.0	4.21	1.67	non-exponential		

earlier. The inherent problem with this technique is that one does not have a good method available to correct for the number of photons scattered from the sample. When curve fitting one must estimate the scattered component before the calculations of the lifetimes. If this estimation is made incorrectly then the added errors will greatly distort the calculated lifetime values.

CHAPTER III

THEORETICAL TREATMENT OF PHASE FLUOROMETRY

In order to improve the collection speed needed to study dyes with low quantum yields or in solutions at low concentrations, it was obvious that another experimental system should be used. Thus, a phase fluorometer was constructed.

A Control Laser Argon Ion Laser Model 556A (25 watt output max.) was used to drive a Spectra Physics Model 375 dye laser. The laser dye used was Rhodamine 6G perchlorate with a tuning frequency of 550 nanometers to 630 nanometers. The output from the dye laser was modulated at exactly 100.1 MHz by an Interaction model SWM-50 acousto-optic modulator. The modulated beam passed through a sample compartment and into a Jobin-Yvon Model 5/367 UV monochromator. The photons were detected by an RCA Model 4084 9 stage side-on photomultiplier and the signal was amplified by a Pacific Precision Model 503L 3 watt radio frequency power linear amplifier. The signal from the amplifier was then input to a Princeton Applied Research Model 5202 Lock-in Amplifier.

With this type of apparatus a simulated phase shift of 30 degrees was used by increasing the path length of the simulated decay measurement by a movable prism. After repeated measurements a precision in the phase angle measurement of 0.25^{o} was observed. At a modulation frequency of 30 MHz this gives a percent relative standard deviation of a simulated lifetime measurement of greater than one percent.

The difficulties in precision and the percent relative standard deviation when using a lock-in amplifier include the precision of measurement, the shielding of cables, and the expense of the equipment. The cost of a good lock-in amplifier can range up to ten thousand dollars. Even if the shielding problems are overcome, repeated measurements show that the precision of the phase angle measurement is typically on the order of a quarter of a degree.

It should also be noted that one problem in using a lock-in amplifier to measure these phase shifts is that the highest modulation frequency that is available on most lock-in amplifiers is 50 MHz. In this connection the phase shift technique can be shown to give the smallest errors in the measurement of a lifetime at an angle of 45°. In the phase shift equation,

$$\tan\phi = 2\pi\nu\tau$$

It can be seen that if the modulation frequency is varied, then the phase shift angle measured must change. The lowest relative error will occur when the phase angle is 45° or rather when the modulation frequency is equal to the inverse of the lifetime. In order to obtain a precision in the lifetime in the picosecond range, the modulation frequency must be above 50 MHz to obtain precision of several percent.

The effect of the phase shift angle upon the percent relative standard deviation can be shown in Table V. For a lifetime measurement of 150 psec the percent RSD for the phase angle shift for a modulation frequency of 30 MHz is only slightly higher than the 100 MHz modulation frequency. Using higher modulation frequencies in the case of very short lifetimes is advantageous in that there will be a larger phase shift in

TABLE V

%RSD VERSUS PHASE MODULATION FREQUENCIES

Phase Shift	
30 MHz 5.31 nsec %RSD = 24.69	100 MHz 1.592 nsec %RSD = 7.41
30 MHz 0.15 psec %RSD = 1.72	100 MHz 0.15 psec %RSD = 1.02

the higher modulation case. As the modulation frequency is increased and the phase angle approaches the ideal 45° phase-angle shift, the minimum percent relative standard deviation will be obtained. To counter this error one might use a delay line or a movable prism in order to increase the path length of the reference beam. If one knows the delay plus the actual lifetime, then one might optimize the measurement by using a calibrated delay to obtain a 45° measurement. If one only observes the error in the precision of the measurement by following the tangent function then one would expect that if the modulation frequency could be determined exactly, then the percent relative standard deviation would be a minimum at a low phase angle shift. Table V indicates that this assumption is the case. For a 45° phase shift the 30 MHz modulation frequency for an error in the phase angle reading of 0.1° will give a percent relative standard deviation of 25 percent while for the 100 MHz modulation the error is 7.5 percent. For no phase shift at 30 MHz the percent relative standard deviation is only 1.72 while at 100 MHz the error is only 1.02 percent. In practice, these percent relative standard deviation values at low phase angle shifts will not be realized due to instabilities in the modulation frequency measurement. This point indicates that for short lifetimes a phase shift angle of 45° is no real advantage when considering that one has a measurement error reading to 0.1°. For the 30 MHz modulation frequency one would have a percent relative standard deviation of almost 25 percent while the 100 MHz modulation frequency gives a 7.5 percent relative standard deviation. In practice, the real precision problem for short lifetimes occurs because one cannot measure the phase angle to better than 0.1°.

When calculating a phase shift using the phase technique the relative phase angle is measured between the reference and sample curve. The phase angle shift of any two sine curves can be calculated from the tangent of the phase angle shift of the reference versus the sample sine wave. This phase angle shift can then be used to calculate the lifetime.

If one describes curve A as the sine of theta, then

$$\cos\theta = \sin(\omega t + \theta)$$

The sine of theta can be found from the integral of the decay curve.

$$\sin\theta = \int_{A/4}^{3A/4} xdx - \int_{3A/4}^{5A/4} xdx$$

Likewise, the cosine of the angle is found by integrating using the following limits:

$$\cos\theta = \int_{o}^{A/2} xdx - \int_{A/2}^{A} xdx$$

The tangent of the angle is just,

$$\frac{\sin\theta}{\cos\theta} = \tan\theta = \frac{(\int_{A/4}^{3A/4} xdx - \int_{3A/4}^{5A/4} xdx)}{(\int_{o}^{A/2} xdx - \int_{A/2}^{A} xdx)} \qquad (4)$$

When the four integrals using Equation 4 are completed the lifetime can be calculated. Alternatively, the lifetime could also be obtained by simply calculating theta by the integration of either the two sine integrals or the two cosine integrals. In practice an advantage was realized by using Equation 4 in that the errors made in the integration of the curves due to background noise tend to cancel out or be reduced because of the division. The data in Table VI gives a real interpretation to simulated errors involving the integral calculations. In case 1

TABLE VI

TANGENT FUNCTION ERRORS

case 1	sin - 0% high	sin	0 % RSD error
26.7 deg	cos - 5% high	cos	17.2%
		tan	14.2%
case 2	sin - 5% low	sin	7.7%
55.4 deg	cos - 0% low	cos	0
		tan	2.15%
case 3	sin - 5% low	sin	7.7%
55.4 deg	cos - 5% low	cos	5.6
		tan	6.27
case 4	sin - 5% high	sin	9.0%
55.4 deg	cos - 5% high	cos	5.6
		tan	6.1

the two cosine integrals produce an integral 5 percent too high, while the error in the integral of the sine is kept at zero. If this error would occur, the error for the lifetime calculated from the cosine phase angle integrals would be greater than 17 percent while the error in the tangent is only 14 percent. In case 2 the sine was estimated at 5 percent too low while the error in the cosine phase angle calculation was zero. The result again shows that the error in the tangent will generally be between the lifetime errors obtained from just calculating the lifetime from the sine or cosine phase shift angle alone. The results in Table VI show that the integral and data collection errors will tend to be reduced in many cases simply by using the tangent function equation.

When one calculates the tangent phase angle, a point on the reference curve is picked as a reference point from which to begin integration. The phase angle shift can be found from this reference point on the reference curve. The phase angle shift is found from this point on the reference curve versus a relative point on the sample curve. The absolute value of the difference of the two relative phase angles was taken as the phase angle shift. The difference phase angle value is related by Equation 4 to the time delay of the photons excited to a higher electronic state in the fluorescent sample. The absolute value of the phase angle can be used to find the lifetime.

$$\theta = \text{ATAN} \left(\int_{A/4}^{3A/4} x dx - \int_{3A/4}^{5A/4} x dx\right) / \left(\int_{o}^{A/2} x dx - \int_{A/2}^{A} x dx\right)$$

To eliminate the precision problems plaguing the phase fluorometer a multichannel analyzer was used as the detection device instead of a lock-in amplifier. The advantage of analyzing the phase shift by counting single photons is that the percent relative error in each channel should decrease with the square root of the reciprocal of the number of photons counted. In other words, the errors in estimating the lifetime of the phase shift will decrease with the length of time that one counts.

Another advantage of single photon counting phase fluorometry in comparison to the single photon counting method is that a modulated laser is used as the source instead of a flash lamp. With a photon arriving at the detector every one hundred nanoseconds one observes that this counting rate is much improved over a flash lamp flashing at a maximum rate of nearly 10 KHz and comparable with modelocked cavity-dumped laser performance.

In the single photon counting technique the background is usually estimated and subtracted from the total number of counts. In many cases this estimation of the background intensity is a visual estimation. This simple visual inspection method in many cases is inadequate and errors can be introduced. For most fluorescent spectral problems it was seen from the lipid membrane experiments in Chapter II that the visual inspection method was inadequate. Large errors were introduced into the measurement of the fluorescent lifetime. However, in the multichannel analyzer arrangement described here these background errors can be almost totally eliminated.

Equations 2 and 3 eliminate the background component by subtraction of the total DC component from the AC component by a difference between the integrals. The subtraction of the integrals is a method to determine the AC component of the data. Each independent integral is a partial summation of the sine or cosine portion plus the background DC component. If the background is truly a random scattering of photons, then it will be represented as a flat curve. The subtraction of the integrals eliminates this background.

The usable portion of the signal that has not been subtracted out can be called the "well depth" of the sinusoidal function. That is, the well depth is the number of counts from the maximum of the sine curve to the minimum of the sine curve. In a typical multichannel counting experiment the number of counts of the well depth will increase with the real length of time of data collection. One would expect that as the well depth of the sine curve is increased the noise level should decrease in proportion to the square root of the number of counts. As this well depth portion of the sine curve is increased the errors due to random noise will decrease. The effect of increasing the well depth can be seen in Table VII.

TABLE VII

% RELATIVE ERROR VS. WELL DEPTH

(2 Runs) 55.6 Degree Shift Well Depth (Counts)	100 MHz Modulation Percent Relative Error
2 K	0.482
4 K	0.377
8 K	0.195

In an effort to simulate the percent relative error that one would anticipate as errors due to counting noise, a computer program was written to simulate the effects. Data curves were synthesized using the function:

$$y = A\sin\theta$$

A random amount of noise was added to the function in Equation 5. The function used was RANDU which is shown in the appendix. The random number generator program uses the power residue method of generating a random decimal number between 0 and 1.0 (57).

Many random functions are determined from an equation:

$$\mu_{i+1} = a\mu_i + c$$

where μ is any random integer and c is the constant. A multiplicative random generator occurs when c equals zero while a mixed random generator occurs when c is not equal to zero (57). Alternate random number generators can be shown to be,

$$\mu_{i+1} = (2^n+3)\mu_L$$

$$V_{k+1} = (2^n+1)V_k+1$$

The last two functions have been shown to give a more random distribution with a larger distribution of random numbers (57). The relative random noise added in each instance was proportional to the square root of the number of counts in each particular channel. In the computer program the function in Equation (5) was shifted by an appropriate phase angle and the decay lifetime was then recalculated. The actual decay lifetime was then compared to the recalculated decay. The percent relative error and the absolute error were calculated from the difference of these two values.

From Table VII it can be seen that the random counting errors at a well depth of 2 K counts ideally produces a percent relative error of less than one percent. As the absolute well depth of the modulated curve is increased the percent relative error falls off. Thus, the error decreases by the square root of the number of counts collected in each respective channel.

Other factors besides the number of counts of well depth that are important are the phase shift angle and the frequency of modulation. Schlag (41) finds that for a pure exponential decay, the error of the phase shift at a constant total count is a minimum at an angle of 45°, which occurs if the modulating frequency is equal to the decay constant. These two factors are inseparable because the modulation frequency can be varied in order to make the phase angle relatively close to 45 degrees.

In Table VIII a decay curve is shown with simulated random noise added. The table indicates that if the phase shift is far away from the idealized 45° phase shift a larger percent relative error will be obtained. One can also observe that as well depth increases the percent relative error drops roughly as the square root of the number of counts. The data presented in Table VIII was synthesized with a 100 MHz modulating frequency and a 69.8° phase shift (approx. 4.33 nsec simulated lifetime). Results shown in the table indicate that with a well depth of 2 K counts in the maximum channel an error of less than one percent should be theoretically possible. Furthermore, the results from Table VIII indicate that a fluorescent molecule that has a lifetime of only 100 psec an error of less than one percent can be obtained with a well depth of only 32 K counts.

TABLE VIII

% RELATIVE ERROR VS WELL DEPTH

Well Depth	4.33 nsec Lifetime 69.8 Degree Shift		100 psec Lifetime 3.59 Degree Shift	
	% RSD	± psec	% RSD	± psec
2 K	0.3755	162.6	8.16	8
4	0.323	139.9	4.71	4
8	0.166	71.9	1.742	2
16	0.0745	32.3	1.198	1
32	0.0355	15.4	0.959	1

A comparison should be made of these values with that of other techniques in the literature. In 1981, Greer, Reed, and Demas studied the accuracy and precision of the phase plane method of deconvolving flash lamp decay curves. Their results indicated that with a simulated flash lamp of 1.4 nsec FWHM they could deconvolve a lifetime as short as 2.5 psec with accuracies and precisions of between 14 and 29 percent (10). In their example, they deconvolved a decay curve of 10 K and 100 K with random noise added. The phase shift lifetime simulated was from 2.5 to 100 psec. For each condition twenty simulated decay curves were generated and deconvolved by the phase plane algorithm. The results are reported as the relative error (RE) and the percent relative standard deviation (% RSD), where

$$RE = \left| 100 \cdot (\bar{\tau}-\tau)/\tau \right|$$

$$RSD = 100 \cdot \sigma/\tau$$

where $\bar{\tau}$ is the simulated lifetime, and τ is the true lifetime, and σ is the standard deviation.

The results of Greer, Reed, and Adamson were compared to the results obtained from the phase shift routine of this work. In each simulated curve computed by the phase shift technique the phase shifted sine curve was computed and random noise equal to the square root of the inverse of the number of counts in each particular channel was added. The phase angle of the simulated run with noise was compared to that of the actual decay.

Initially it appears that much superior values were obtained from the phase-plane method of deconvolution over the phase angle shift technique. Further study of the two techniques shows that the slight advantage of the phase-plane method disappears at longer lifetimes. When the lifetimes are over 50 psec the percent relative standard deviation is nearly the same for 10 K and 100 K count maximums. However, at lower lifetimes the single photon counting phase angle shift method appears to have a higher percent relative standard deviation. In analyzing the percent relative standard deviation one should realize that the phase shift simulation was run using a modulation frequency of 100 MHz. If the modulation frequency were increased then the errors using the phase shift simulation technique would drop until the optimum phase angle shift of 45° were obtained.

The errors in the phase angle shift method can be reduced further by using a typical multichannel analyzer with 1024 channels. The phase angle data presented in Tables IX and X is integrated over only one

TABLE IX

SIMULATION CURVES-10 K MAX. COUNTS WELL DEPTH

Shift (nsec)	Phase Plane		Phase Angle Shift Method	
	%RE	%RSD	%RE	%RSD
2.5	8.2	28.4	13.5	78.76
5.0	6.1	14.8	6.7	42.17
10.0	1.5	8.5	3.5	21.71
25.0	0.0	6.7	1.4	8.87
50.0	0.6	3.6	0.67	4.45
100.0	0.3	2.4	0.199	2.21

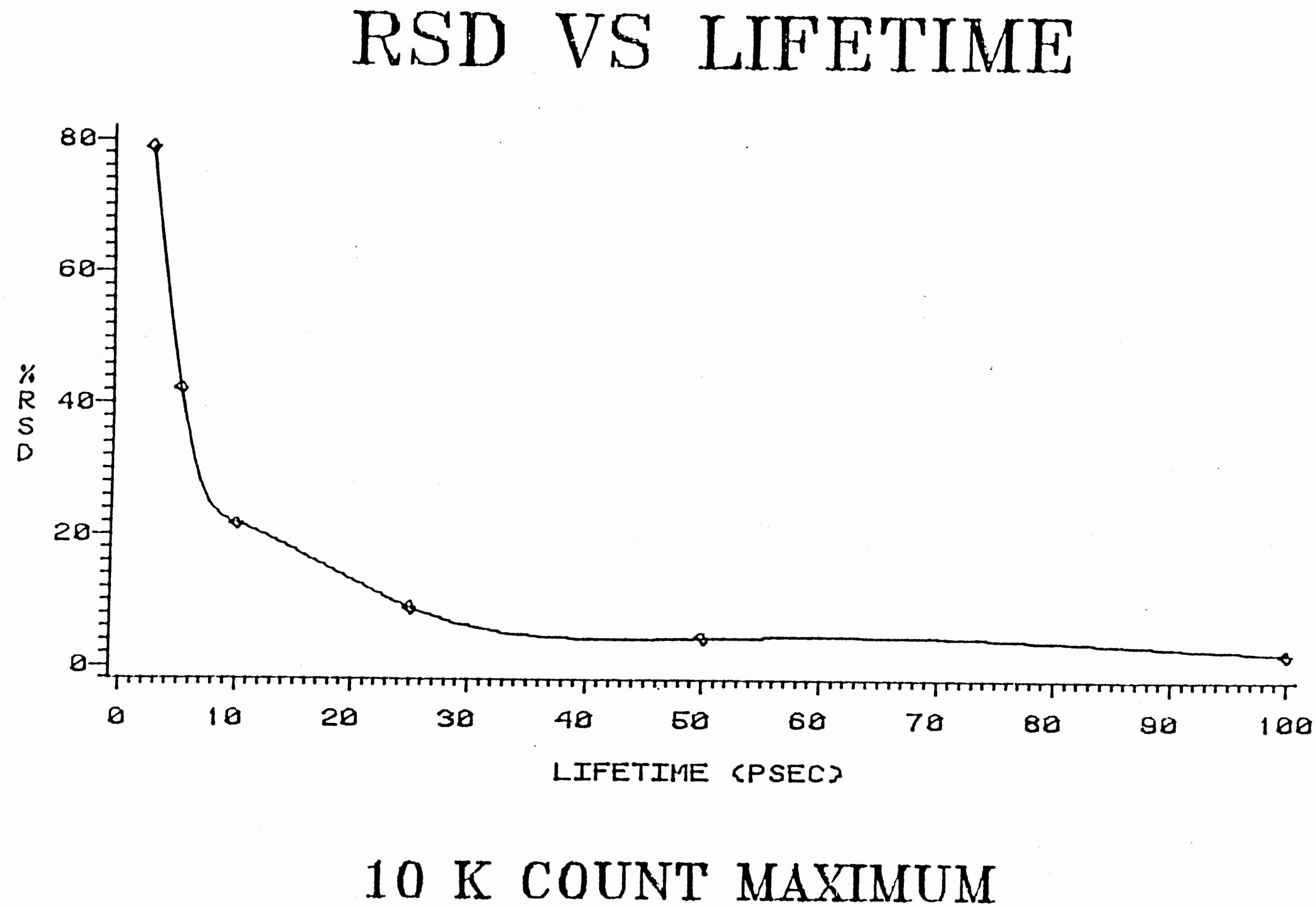

Figure 8. Percent Relative Standard Deviation Versus the Fluorescent Lifetime Measurement - 10K Count Maximum

TABLE X

SIMULATION CURVES-100 K COUNTS MAX. COUNTS WELL DEPTH

Lifetime (nsec)	Phase Plane		Phase Angle Shift Method	
	%RE	%RSD	%RE	%RSD
2.5	0.4	13.7	4.9	25.2
5.0	1.6	6.6	2.5	12.9
10.0	0.6	2.4	1.3	6.5
25.0	0.9	3.1	0.49	2.6
50.0	0.0	1.5	0.21	1.3
100.0	0.2	0.5	0.02	0.649

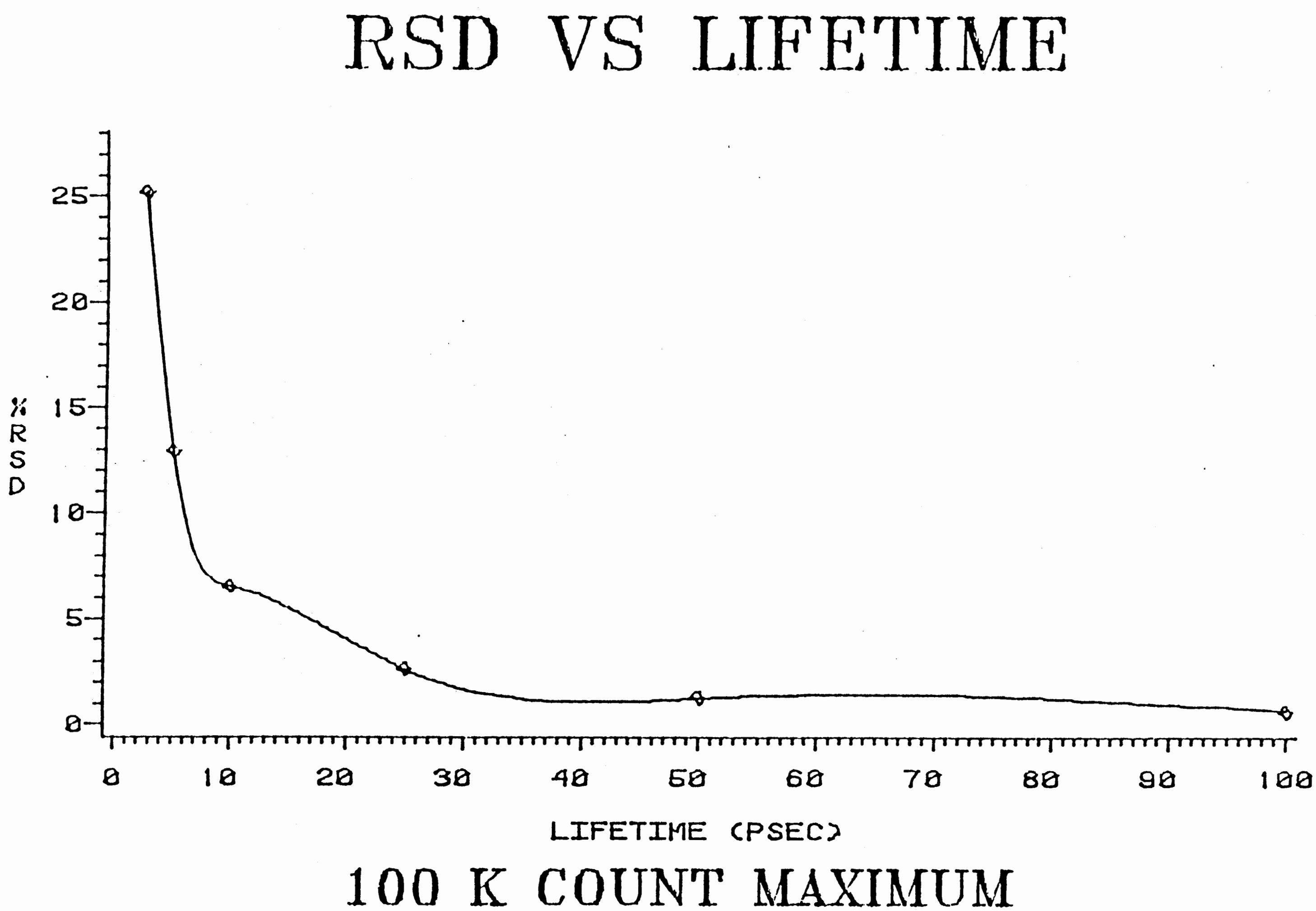

Figure 9. Percent Relative Standard Deviation Versus the Fluorescent Lifetime Measurement - 100K Count Maximum

sinusoidal wave period. If one averages over several periods the errors can be reduced. The phase angle method has an advantage over the single photon counting method in that several periods can be averaged giving a lifetime value. In the single photon counting method only one value is obtained from a curve fitting routine.

It should be noted in Tables IX and X in the data by Greer, Demas, and Adamson that the errors for a 10 picosecond shift were lower than the errors for a 25 picosecond shift for the 100 K count maximum. These results seem to be in error considering that twenty runs were made in the average calculation of the percent relative standard deviation. The errors should be lower for longer decay lifetimes.

In their calculations Greer, Demas, and Adamson use a Hewlett-Packard 9825A desk top calculator to simulate their results. The programs were written in BASIC programming language and used the Hewlett-Packard systems 0 to 1 random number generator. The possible error in the results of the phase plane method may occur in the random number generator. Errors can occur in the way random number generators are used in the program to add in the random noise. In particular, difficulties can occur in generating random numbers if the input integer is not always sufficiently random. If the integers are chosen in a strictly deterministic fashion, then the distribution of values will not be wide enough.

The phase angle shift method is believed to give good results basically because of two reasons. First, the phase angle measurement is devoid of high background stray light problems. This fact may be especially important for experiments in which the fluorescent species of interest is low in concentration or in intensity. Secondly, the phase angle value is determined from the total curve. Since the calculation of the phase angle uses the integral of the total curve, a point which is higher in intensity is necessarily given more relative weight. This condition is favorable because the channel with the higher number of counts should have a lower relative percent of random noise. In the single photon counting method the whole curve is analyzed, but much of the exponential decay curve information used in the fitting routines is actually obtained from the portion of the curve that is lower in intensity. The data taken from the decay curve at this point will have a higher relative percent of stray light to be subtracted then the phase shift method. Because of these two reasons the single photon counting phase angle shift method should generally result in measured lifetimes of very good precision.

CHAPTER IV

EXPERIMENTAL RESULTS

With the theoretical calculations and simulations indicating that precision measurements in the picosecond range could be obtained, an experimental apparatus was built. The purpose of the experimental conditions were to confirm the theoretical results. The primary objective of the experimental construction was to build a photon counting phase fluorometer with commercial subunits with emphasis on ease of construction.

The experimental apparatus used a Hughes Model 3225H-P2 10 mw He-Ne laser. The 633 nm output was passed through an Interaction Acousto-optic Modulator SWM-50. The modulator was driven by the RF output of an Interaction Model ME-40 Signal Processor operating at 50.05 MHz. The 633 nm beam was taken at the Bragg diffraction angle to give a modulated signal of 100.1 MHz. The amplitude modulated frequency of the laser beam was at two times the modulating frequency of the acousto-optic crystal driver.

From the exit slit of the monochromator, the modulated light beam enters the sample chamber. Photons emitted by the sample travel through a Jobin-Yvon HR-20 monochromator and are detected by an RCA 4084 9 stage side-on photomultiplier tube. The output of the photomultiplier was amplified by a Pacific Photometric preamplifier. The time delayed pulses were counted by a Lecroy Multichannel Model 3001 Analyzer. The output of the multichannel analyzer was displayed on an NRS Model MS-215 Miniscope oscilloscope. The data was stored and analyzed by an Apple II 48 K microcomputer. The Apple was interfaced via a 300 baud modem to an IBM 370/168 Model 3 main frame computer. The IBM 370/168 was essentially used for its data base management system for storing many data files. The interface was accomplished through a time sharing system on the IBM 370 and the use of a Novation Series 4000 300 baud acoustic modem on the Apple microcomputer.

To begin the experiments the travel time of the photons transversing through the sample and the instrument is measured. The scattering solution that was used successfully for the reference was a whole milk sol diluted by a two to one dilution with water. Once the reference travel time phase curve had been obtained a sample curve could be collected. The phase difference between the sample and the reference curve which was a summation of individual photon arrival times were related to the actual lifetime of the sample.

In determining the phase angle shift, Equation 4 of Chapter III was used. The calculations were accomplished by the use of the Apple BASIC language program SHIFT1 and also the FORTRAN program PHASEL. Both programs calculate the phase angle and the lifetime of the excited state species. The program SHIFT1 used a triangular integral routine to integrate the curves, while the FORTRAN program used a Simpson's rule integral routine in order to integrate the curve.

The data in Table XI shows that both programs give values that are within 0.2% of each other. The differences lie in the fact that the Simpson's Integration is superior in interpolation of the integral between the adjacent data points.

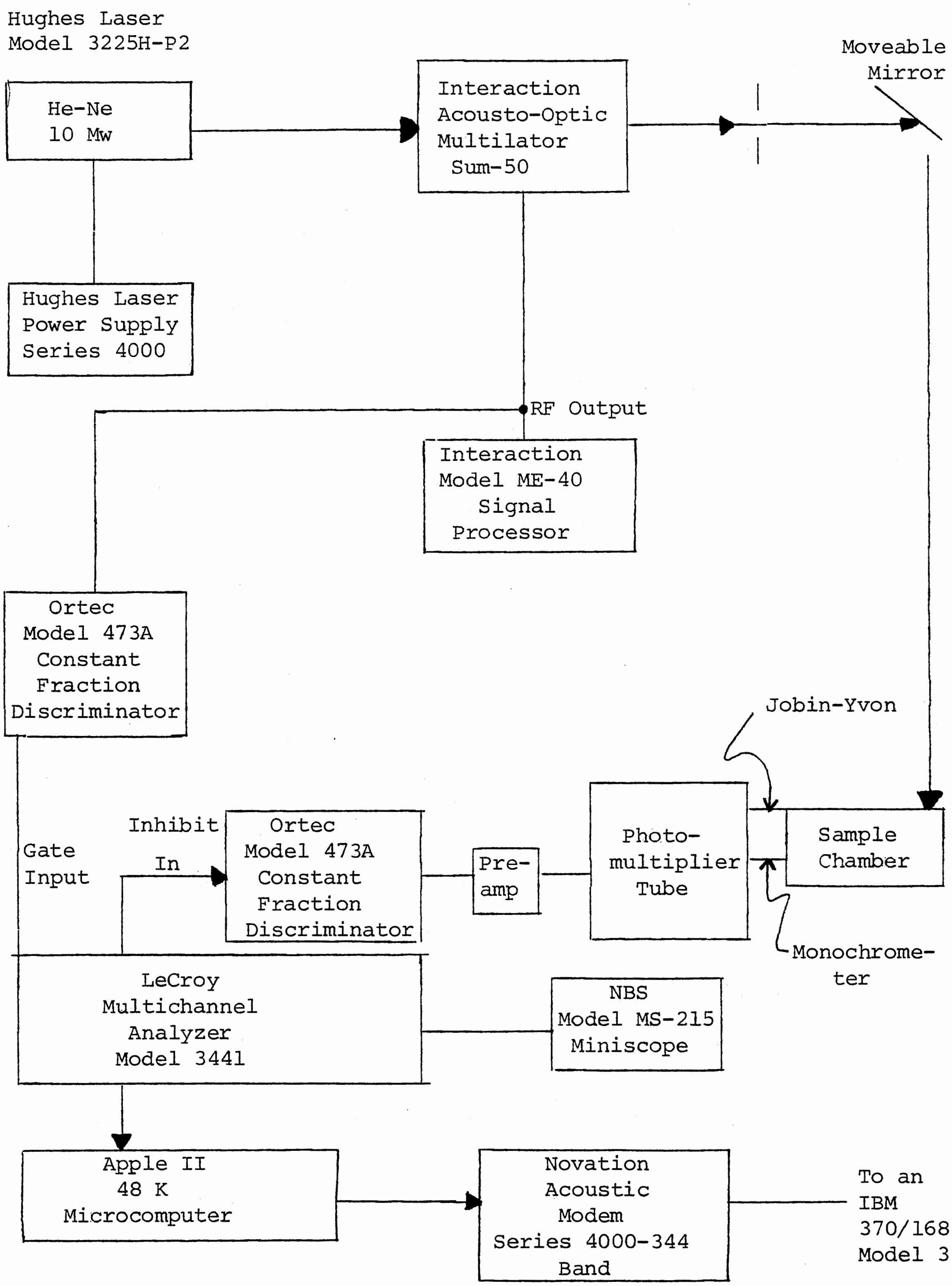

Figure 10. Optical Diagram

TABLE XI

PHASE SHIFT COMPUTER PROGRAMS
ACTUAL LIFETIME = 1.3912 NSEC

Run File No.	Shiftl	PhaseL
1	1.40716	1.4059
2	1.4115	1.4096
3	1.4023	1.4002
4	1.4046	1.4023
5	1.4006	1.3989
6	1.3846	1.3827
7	1.4044	1.4018
8	1.3798	1.3778

The phase shift calibration of the instrument was checked by altering the path length of the beam by a known distance. The optimal delay path was achieved through the use of a calibrated optical slide onto which was attached a movable prism. The accuracy and also precision was approximately 0.1 mm. After the path length was varied a phase shift could be calculated due to the relative delay of photons traveling the path distance. The accuracy and precision of the experimental lifetime values could be checked by the theoretical computer programs. It was hoped that the theoretical lifetime precision values would be matched by the experimental precision results.

A theoretical delay was calculated by simulating a decay curve by an eight channel shift on the multichannel analyzer. Since the precision of the lifetime value is a function of the well depth of the modulated count maximum, the theoretical data was altered by the well depth and background conditions. The well depth used in the experimental calculation was set at 8 K counts with 500 counts of background. Eight theoretical runs were made to simulate the 0.8441 nsec lifetime. The results are shown in Table XII.

The relative standard deviation error for a lifetime of 0.84 nsec with an 8 K well depth should theoretically be around 0.5 percent. The results should be obtained after an initial "warm-up" period of the instrumentation. Lifetime measurements were taken every few minutes to determine the stabilization point after which data measurements could be taken. The results tabulated in Table XII and XIII are for three decay times. The experimental results are shown in Table XIV. The true lifetime should be in the range of 0.866 nsec.

The data in Table XII and XIII indicate an experimental standard

TABLE XII

EXPERIMENTAL LIFETIMES (0.8441 NSEC)

Run No.	Phase Angle (Degrees)	Lifetime (nsec)
1	27.90	0.84268
2	27.87	0.84165
3	27.94	0.84411
4	27.79	0.83877
5	27.98	0.84553
6	28.00	0.84624
7	27.92	0.84339
8	28.03	0.8473

Standard Deviation = 0.00406 nsec or 0.48% relative standard deviation.

TABLE XIII

8 K WELL DEPTH

Run No.	Lifetime (nsec)
1	0.81749
2	0.83368
3	0.82416

Standard Deviation = 0.00634 nsec or 0.769 % relative standard deviation.

deviation of about 6 picoseconds with a lifetime of 0.82 nsec. At an 8 K well depth this gives a precision value of less than one percent. The theoretical value of 4 psec for the precision compares favorably to the 6 psec experimental value. The difference may possibly be due to a non-random drift of the instrumentation caused by "warm-up" hysteresis or simply due to the fact that the count rate of the photomultiplier changed.

In a study of the precision values obtained; several non-uniformities such as the "warm-up" hysteresis were found in the data. The areas that were deemed important for study were the problem of non-uniform peaks, fluctuations in the rate of photons arriving at the photomultiplier tube, and the hysteresis effect of the photomultiplier tube during an initial "warm-up" period.

In the previously mentioned study of the "warm-up" interval required for instrumental hysteresis of the photomultiplier tube, twelve runs were made after the photomultiplier was turned on. A moveable mirror was shifted 18.8 cm which corresponds to a lifetime of 0.8 nsec. The maximum well depth of the experimental decay curves was 6 K. The results of the lifetimes are shown in Table XIV.

The data from Table XIV shows that the lifetime fluctuates badly up to 50 minutes. This indicates the "warm-up" time required for all the electronic fluctuations to drop. The hysteresis that is seen is probably due to the initial instability in the anode current. This change in the anode current causes a change in the charging of the dynode support spaces. This causes a change in the electron optics in the tube and a resulting change in the transit time of the electrons cascading through the tube.

TABLE XIV

WARM-UP PERIOD OF THE PHOTOMULTIPLIER

Run No.	Time From Turn on (Min.)	Lifetime (nsec)
1	10	0.9065
2	15	0.9974
3	20	0.7269
4	25	0.5211
5	30	0.6660
6	35	0.5770
7	40	0.6170
8	45	0.7448
9	50	0.7491
10	55	0.7569
11	60	0.7669
12	65	0.7673
13	70	0.7684
14	75	0.7543
15	80	0.7580
16	85	0.7712
17	90	0.7761
18	95	0.7661
19	100	0.7650

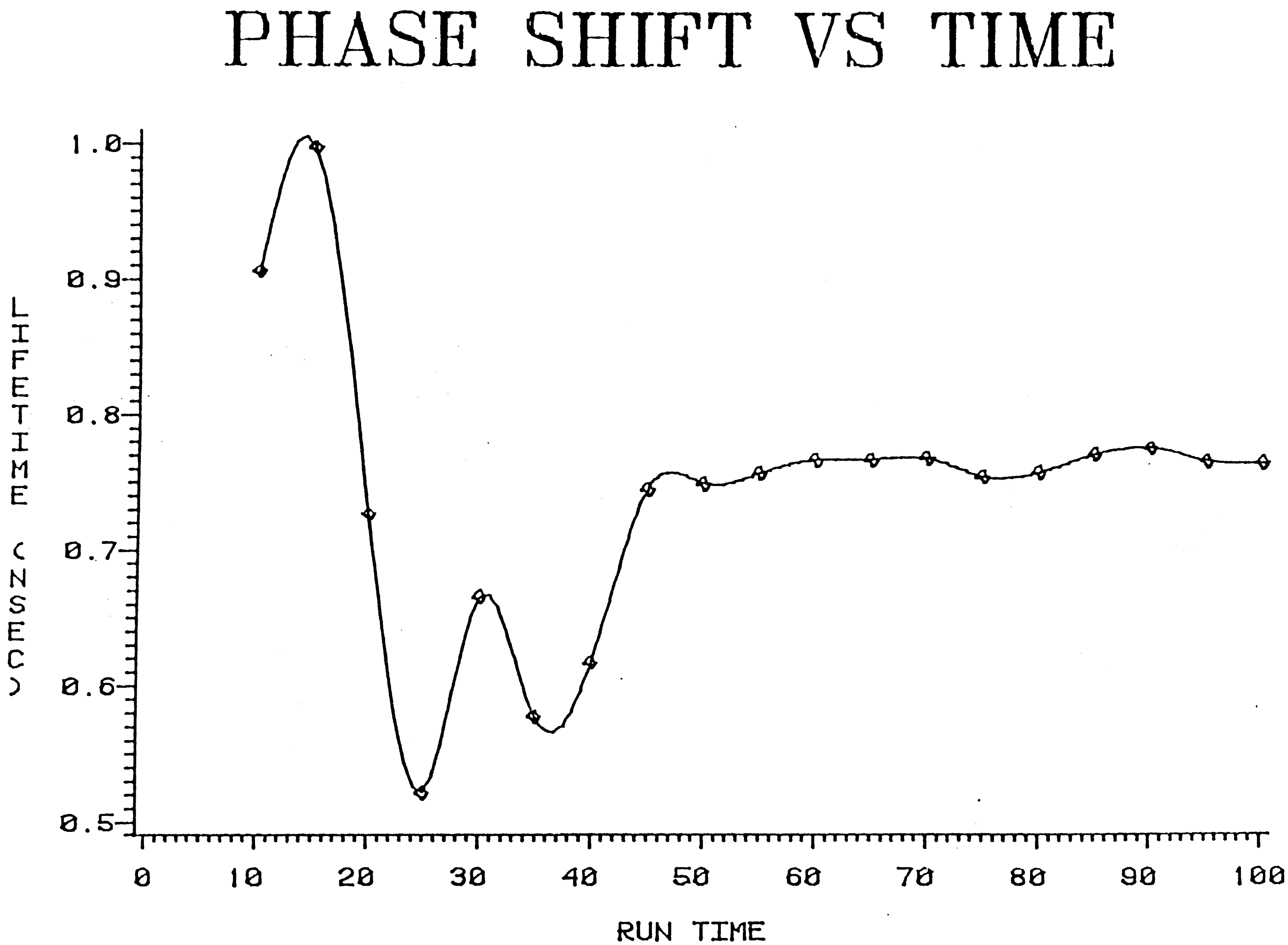

Figure 11. Phase Shift Lifetime Versus Real Run Time

Superimposed oscillations as described below, can be a very serious error in estimating the correct lifetime. This is because the phase angle is determined by an integral of the decay curve. If the decay curves are not uniform because of background noise pick-up from lower frequency sources then one would expect that the lifetime values calculated would be smaller than is actually the case.

After isolating each device in the line it was found that most of the non-uniformity in the peaks of the decay curve originated from pick-up by the photomultiplier leads. The coherent pick-up originated from the 50 MHz acousto-optic crystal power supply driver. The result was a superimposed 100 MHz phase shifted curve with a 50 MHz signal from the driver that was not phase shifted. This effect of the non-shifted 50 MHz oscillation was to reduce the actual lifetime value obtained and destroy the accuracy of the method. Thus, the absolute lifetime value calculated was dependent on the percentage of the lower frequency oscillations received. For very accurate lifetime values it can be seen that the photomultiplier lines to the multichannel analyzer must be isolated.

To indicate the serious nature of lower frequency oscillations in real data, a 50 MHz signal was superimposed over the 100 MHz signal. The percentage of the 50 MHz signal was 5 % of the total well depth. With just 5 % of the 50 MHz oscillation the lifetimes are shown to be consistently much smaller than the true value. Table XV shows the results of an experimental delay line simulation. The results indicate that as the phase angle shift approaches an angle of 45° the minimum percent deviation from the true measured results is approached. Thus, even if the low frequency signal is present lower errors should be obtained at a phase shift angle of 45°.

TABLE XV

DEGREE SHIFT VS % ERROR

Run No.	Shift	(cm)	Simulated Delay (nsec)	Cal. (nsec)	% Error
1	0	0	0	36 psec	100.0
2	5	12	0.338	0.1899	43.8
3	10	24	0.708	0.495	30.1
4	15	36	1.155	1.108	4.1
5	20	48	1.766	1.35	23.5
6	25	60	2.75	2.165	21.3
7	30	72	4.89	2.796	42.8

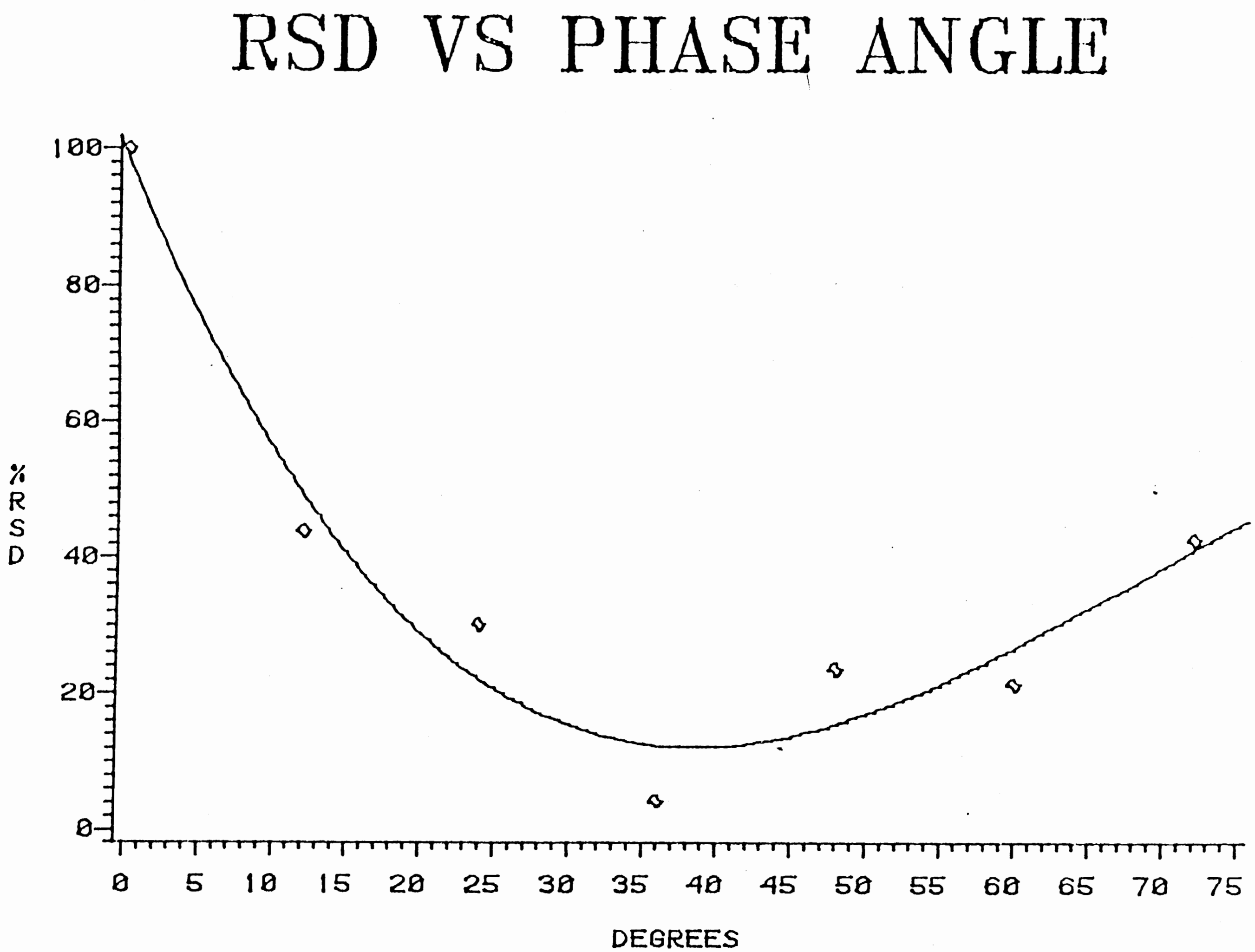

Figure 12. Experimental Percent Relative Standard Deviation Versus the Phase Angle Measurement

This reduction in the measurement error for the phase angle shift is typically what one would expect from experimental data. As the phase angle deviates from a 45° shift, added measurement errors are encountered. The explanation for this phenomenon lies in the definition of the tangent function. If the tangent function is plotted from -90 to $+90^{\circ}$ and the y axis is a scale of $2\ \pi\nu\tau$, then it can be seen that the 45° shift is a very linear portion of the curve. From an instrumental viewpoint, great errors in resolving the lifetime, τ, would be read if the phase shift approximated 90°. Near a 90° phase shift the y axis varies greatly with little variance of the x axis. If a phase shift approximating zero degrees is read one may initially assume that the phase shift errors would be very small. However, if one has a small error in the measurement of the frequency of the modulated source, a larger error in the calculation of the actual lifetime value may result.

Another possible source of error in lifetime measurements is due to the count rate fluctuations at the photomultiplier. As the count rate is changed the transit time of the photons varies remarkably. Because of the ability of this instrument to measure transit time intervals precisely it is apparent that this apparatus could be used.

To monitor the lifetime fluctuation versus the count rate fluctuation, several neutral density filters with varying optical densities were examined. In each experimental run two neutral density filters were used so that small differences in the count rate could be achieved by using a combination of the filters. No other setting was changed on any of the instrumentation; only the filters were changed to reduce the intensity of the modulated beam. The results shown in Figure 7 show that the lifetime calculated from the phase shift is altered by the

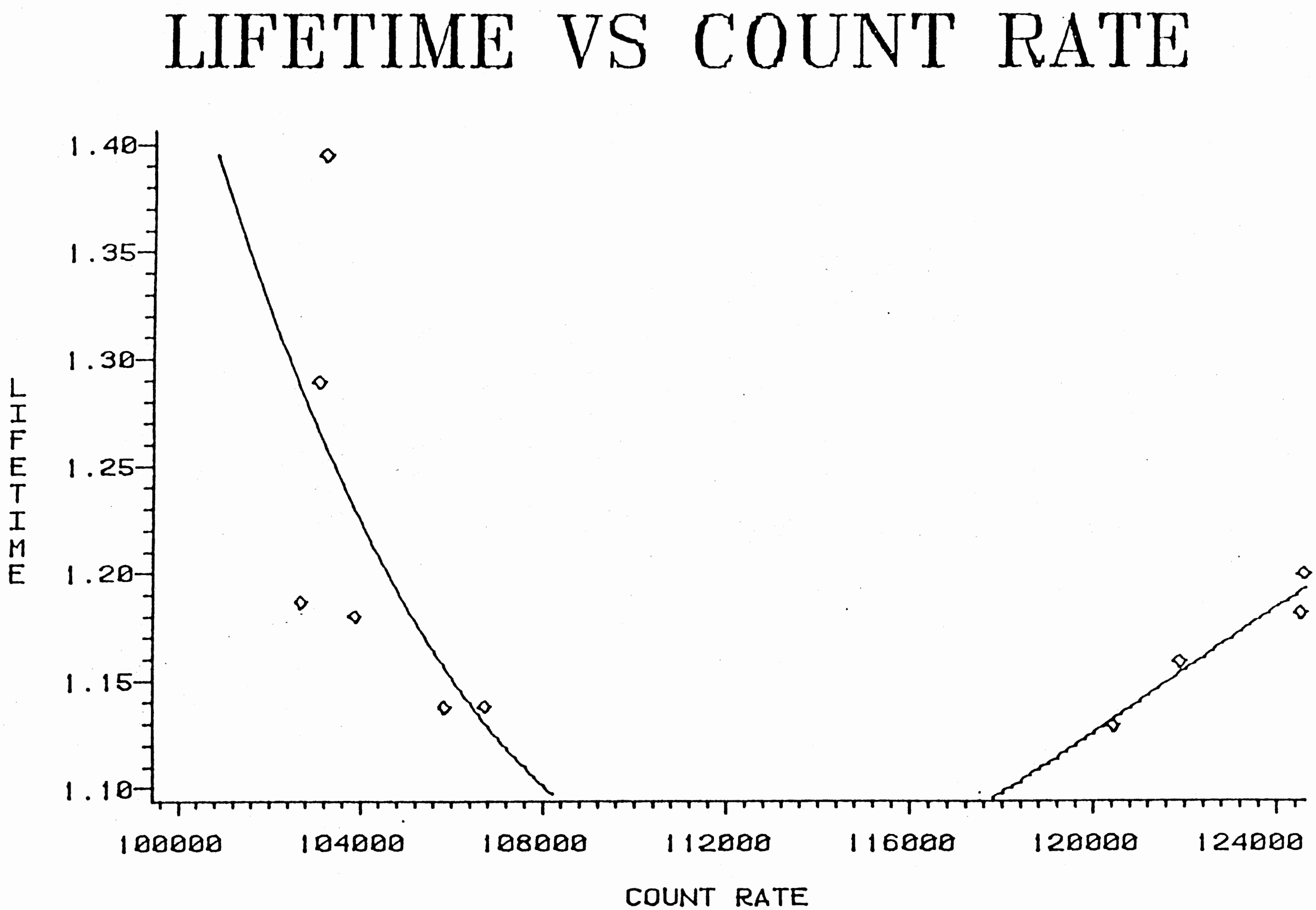

Figure 13. Lifetime Measurement Versus the Photomultiplier Photon Count Rate

change in time it takes the photons to be detected by the photomultiplier. The figure indicates that the calculated lifetime varied 300 psec with a change in the count rate of 24 K counts/second. From these results it is apparent from a linear extrapolation that if the count rate fluctuation could be held to less than 150 counts/second then a precision of roughly 2 psec could be achieved. After repeated efforts it was realized that a count rate fluctuation of less than 150 counts/second could be obtained if one used a neutral density filter wedge. A neutral wedge mounting device was built with the option of moving any part of the wedge into the path of the beam. The effect of the wedge was to give the ability to "fine tune" the count rate of photons entering the photomultiplier. With this adjustable wedge all count rates could be held within a variation of about 100 counts/second. This value was more than adequate for the 2 picosecond limit of precision.

After elimination of each error, corrected experimental optical delay line measurements were taken. A moveable prism was aligned so that prism movements of 0.1 mm could be seen. The measurements consisted of running a reference profile and then running the shifted decay profile. The prism was shifted in a random pattern in 5 centimeter intervals. The experimental lifetimes that were obtained agreed well and were close to the experimental error of the expected lifetime.

The results in Table XVI show that the percent relative standard deviations of each optical delay line measurement were within the standard deviations of the prism movements. In other words, it is believed that the actual limitation in obtaining phase shift values in this experimental configuration was the precision with which one could control the path length of the modulated beam! The data indicates

TABLE XVI

EXPERIMENTAL DELAY LINE MEASUREMENTS (10 K COUNTS)

Shift (cm)	Expected Lifetime	Experimental Lifetime	Error	% Error
0	0 nsec	2.59 psec	2.59 psec	----
5	0.338	0.336 nsec	2.00	0.59
10	0.707	0.704	3.00	0.42
15	1.155	1.108	47.00	4.05
20	1.766	1.802	36.00	2.04
25	2.75	2.774	24.00	0.87
25	2.75	2.627	123.00	4.47
30	4.89	4.958	68.00	1.39
30	4.89	4.882	8.00	0.16

that without any phase shifting an experimental lifetime precision of 2.5 psec may be experimentally achievable for a well depth of 10 K counts and a lifetime of 0.338 nsec.

The tremendous advantage of this technique is evident when it is realized that the well depth of any of the runs was only 10 K counts in the maximum channel. A precision of 2.6 psec at 100 MHz modulation and 10 K counts of well depth is typical. If one had a well depth of 100 K counts and a modulation frequency above 100 MHz then one might easily measure sub-picosecond events.

The utility of resolution of lifetimes in the picosecond range can no better be displayed than by the study of the formation of aromatic organic excimers. An excimer is produced by a collisional interaction of excited and unexcited molecules to produce an excited dimer. The excimer is not a normal dimer, but is actually a dimer in the excited state and a dissociated molecule in the ground state. Many organic excimers have been found at room temperature in various solvents (56).

From the rate process diagram the monomer is excited into the excited state $^1M^*$ from which some molecules dimerize to the $^1D^*$ state. In this process there are two radiative pathways of decay; monomer emission and excimer emission. In most cases these lifetimes are only slightly different. The method of detection of excimers is to increase the molar concentration which will increase the probability of dimer formation. If one can monitor the lifetime as a function of increasing concentration, then one should be able to see an increasing component of the dimer in the fluorescent lifetime.

The solution chosen here for study was Nile Blue A because it could be excited by a 633 nm He-Ne 10 Mw laser (58, 59). The absorp-

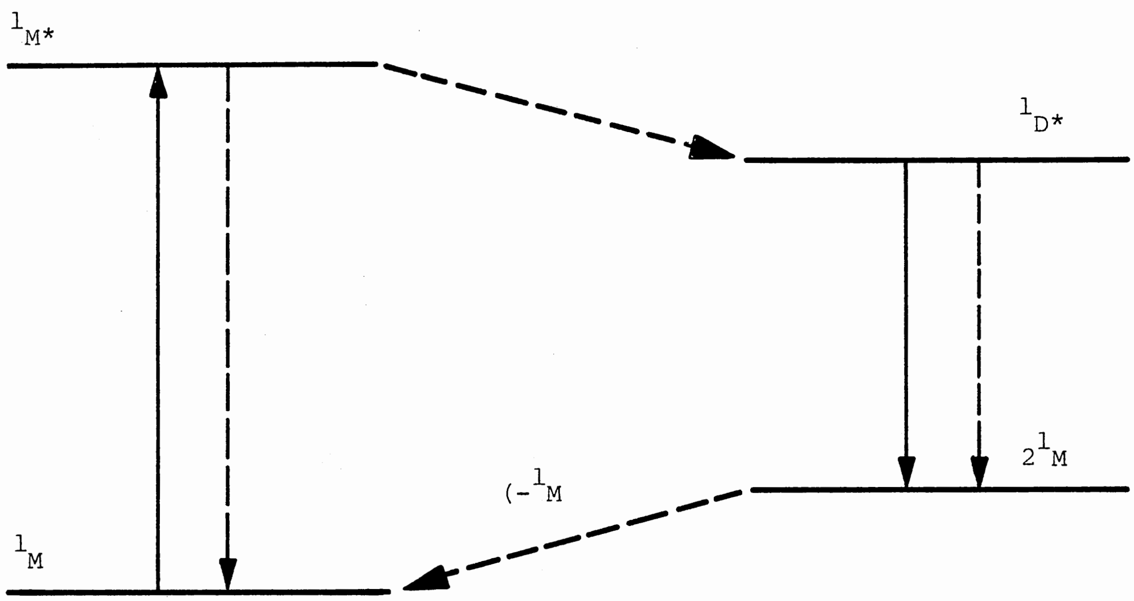

Figure 14. Rate Process Diagram

tion maximum for Nile Blue A is 630 nm with its emission maximum at 660 nm. Several solutions of various absorbances were made. All solutions were run in a non-consecutive manner in order to eliminate any biasing that might occur in the data. Data collection runs were made five times for each solution and the shifted curves were first compared to a reference curve and then to a data file of a sample of the Nile Blue A solution with the highest optical density. The reference solution used was a milk solution that was collected with a well depth of 2 K in the maximum channel. The decay profiles were all collected in a few minutes with a maximum well depth of 2 K. From the data presented in Table XVII it can be seen that a distinct relationship exists between the absorbance and the lifetime of the dye.

The results in Table XVIII present the phase shift lifetime with respect to the optical density of the solution. All lifetime values have a rather high standard deviation. This can be explained because the maximum well depth on any of the runs was only 2 K.

From the figure it is shown that as the absorbance is increased the lifetime increases as a smooth function of the concentration. This effect may be due to an excimer formation of the Nile Blue A. The association of a diffusion controlled process in which occurs in fluid solutions is thought to lead to a delayed molecular excimer fluorescence (1).

Other modulation frequencies must be used to determine the transform of the decay function. However, these results are similar to those of Hauser and Heidt (46). Their work showed how a lifetime measurement can be used to interpret the variation of the lifetime of a fluorescent species in which the acid concentration was varied. The rate constant of dissociation and association was determined by varying the pH

TABLE XVII

LIFETIME SHIFT

Solution	Opt Density	Run Vs Ref	Lifetime Vs Decay File (nsec)	(psec)
Solution 1	0.326	1	1.449	--
		2	1.3836	25.86
		3	1.3936	13.47
		4	1.3628	33.81
		5	1.3257	50.71
Solution 2	0.120	1	1.1376	154.13
		2	1.1040	171.69
		3	1.1372	162.58
		4	1.0694	212.51
		5	1.0302	231.74
Solution 3	0.055	1	0.8504	357.24
		2	0.9032	336.26
		3	0.8740	362.59
		4	0.9001	341.74
		5	0.8779	371.33
Solution 4	0.186	1	1.1712	133.88
		2	1.1894	138.17
		3	1.2244	104.59
		4	1.2570	90.34
		5	1.1267	116.45
Solution 5	0.104	1	1.0496	227.63
		2	1.0907	197.59
		3	1.1190	190.22
		4	1.1350	165.75
		5	1.1040	195.37

TABLE XVIII

AVERAGE LIFETIMES

Solution	Optical Density	Lifetime Vs Ref	Lifetime Vs Decay
		(nsec)	(nsec)
1	0.326	1.3829 ± 0.045	24.77 ± 19.35
2	0.120	1.095 ± 0.046	186.53 ± 33.17
3	0.055	0.881 ± 0.022	353.83 ± 14.57
4	0.186	1.212 ± 0.033	116.69 ± 19.98
5	0.104	1.099 ± 0.033	195.31 ± 22.08

of the solution.

The ease of measurement of excimer lifetimes with this instrument should be greatly enhanced. At a 16 K count well depth the precisions obtained for most dyes having a nanosecond lifetime will be several picoseconds.

CHAPTER V

CONCLUSIONS

The advantages of using photon counting phase fluorometry for picosecond events are numerous. The high precision of nearly 2 psec using a multichannel analyzer channel width of 97 picoseconds is good. With a multichannel analyzer that has a smaller channel width lower precision values should be obtainable. Another advantage includes the fact that the precision rises with the number of counts of well depth. In the collection of experimental profiles during real time operation the precision can be predicted from the number of counts measured. An advantage of this method over phase fluorometry is that the signals are analyzed digitally rather than by analog methods. In many multichannel analyzers in which the modulation frequency is 100 MHz or greater several oscillations will be able to be recorded. The averaging over several periods should increase the precision of the measurements.

Two advantages of the technique over single photon counting are that the measurements can generally be made more rapidly and that one can have real time processing of the data. The first advantage is possible because one can use a modulated laser output which delivers a steady flux of photons to the sample chamber. This provides a high duty factor for efficient data collection. The second advantage is the real time data processing. Several interpreted BASIC language routines were written on an APPLE II microcomputer to calculate the lifetimes. The program SHIFT1 can be run in about two minutes which for a single exponential decay fit is faster in real time than even the phase-plane technique. Finally, the method is comparably priced to that of phase fluorometry and single photon counting. The advantage here is that most sub-units are commercially available.

The photon counting phase fluorometer will provide a tool for studying concentration quenching of molecular fluorescence. One will be able to obtain the individual rate parameters from the fluorescent lifetime measurements. The ability to detect small shifts due to excimer formation will provide much information about the possibilities of excimer formation.

The limits of this method will enable one to obtain better precision than is now currently available. Resolutions of lifetimes of less than 2 psec can be obtained with well depths of less than 20 K and modulation frequencies of less than 100 MHz.

In the effort to increase the resolution of the lifetime values calculated one should obtain well depths greater than 20 K counts. With large well depths one may be able to easily drop the limits of the measurement by another order of magnitude. The drop in magnitude might also require a multichannel analyzer with a smaller channel width and a higher modulating frequency. The data collection time would also be longer for the greater well depths collected.

To improve the resolution of the instrument a new multichannel analyzer is currently being installed. The new Canberra Model 481 time-to-amplitude converter gives a time between channels resolution of 50 picoseconds. This resolution is a factor of two better than the LeCroy multichannel analyzer that was used in this series of experiments. The theoretical predictions of the relative standard deviations for a lifetime with a 45° phase shift using a multichannel analyzer with a channel width of 50 picoseconds is 0.1 percent. This value is better than the percent RSD for the channel width of 0.097 nanoseconds.

When a fluorescent emitter of low intensity is present care must be taken so that a low percentage of scattered light is seen by the photomultiplier. The scattered photons with a zero degree phase angle shift that are collected will cause a preferential shift of the phase angle to lower values. Thus, in this arrangement scattered light can lower the true phase angle shift while the stray light effects can be easily subtracted out.

In conclusion, this method combines the best points of single photon counting with pulse techniques to offer a very precise analysis. With the use of a multichannel analyzer with better resolution and the advent of acousto-optic crystals with modulation frequencies of up to 250 MHz, a very bright picture seems to be in store for fluorescent transient lifetime measurements by this method.

BIBLIOGRAPHY

(1) Birks, J. B., in Progress in Reaction Kinetics (G. Porter, ed.), Vol. 4, Pergamon, New York, 1967, p. 239.

(2) Birks, J. B., Photophysics of Aromatic Molecules, Wiley, London, 1979, 680 pg.

(3) Ware, W. R., Baldwin, B. A., J. Chem. Phys. 40, 1703 (1963).

(4) Barltrop, J. A., Coyle, J. D., Principles of Photochemistry, Wiley, London, 1978, 213 p.

(5) Kauzmann, W., Quantum Chemistry, Academic Press, New York, 1957, 354 p.

(6) Brody, S. S., Rev. Sci. Instr., 28, 1021 (1957).

(7) Ware, W. R., in Creation and Detection of the Excited State (A. A. Lamola, ed.), Marcel Dekker, New York, 1971, pg. 213.

(8) Bennett, R. G., Rev. Sci. Instr., 31, 1275 (1960).

(9) Berlman, I. B., Sleingraber, O. J., Benson, M. J., Rev. Sci. Instr., 39, 54 (1968).

(10) Greer, J. M., Reed, F. W., Demas, J. N., Anal. Chem., 53, 710 (1981).

(11) Knight, A. E. W., Selinger, B. K., Spectrochem. Acta., Sect. A, 27, 1223 (1971).

(12) Demas, J. N., Crosby, G. A., Anal. Chem., 42, 1010 (1970).

(13) Phillips, D. L., J. Ass. Comp. Mach., 9, 89 (1962).

(14) Strand, O. N., Westwater, E. R., J. Ass. Comp. Mach., 15, 100 (1968).

(15) Hunt, B. R., Math Biosci., 10, 215 (1971).

(16) Demas, J. N., Adamson, A. W., J. Phys. Chem., 75 (16), 2463 (1971).

(17) Birks, J. B., Dyson, D. J., Munro, I. H., Prog. Roy. Soc., 275, 575 (1963).

(18) Gafni, A., Modlin, R. L., Brand, L., Biophys. J., 15, 263 (1975).

(19) Hundley, L., Coburn, T., Garwin, E., Stryer, L., Rev. Sci. Instr., 38, 488 (1967).

(20) Ware, W. R., Doemeny, L. J., Nemzek, T. L., J. Phys. Chem., 77 (17), 2038 (1973).

(21) McKinnon, A. E., Szabo, A. G., Miller, D. R., J. Phys. Chem., 81, 1564 (1977).

(22) Grinvald, A., Steinburg, I., Anal. Biochem., 59, 583 (1974).

(23) Birch, D. J. S., Imhoff, R. E., J. Phys. E.: Sci. Instr., 10, 1044 (1977).

(24) Marquardt, D. W., J. Soc. Industr. Appl. Math., 11, 431 (1963).

(25) Isenberg, I., Dyson, R. D., Biophy. J., 9, 1337 (1969).

(26) Schuyler, R., Isenburg, I., Rev. Sci. Instr., 42, 813 (1971).

(27) Schuyler, R., Isenburg, I., Dyson, R. D., Photochem. Photobiol., 15, 395 (1972).

(28) Gaviola, E., Z. Physik., 81, 681 (1926).

(29) Gaviola, E., Ann. Physik., 81, 681 (1926).

(30) Gaviola, E., Z. Physik., 42, 853 (1927).

(31) Spencer, R. D., Thesis: Fluorescence Lifetimes: Theory, Instrumentation, and Application of Nanosecond Fluorometry (University of Illinois, Urbana, 1970).

(32) Hecht, E., Zajac, A., Optics, Addison-Wesley, Reading, Massachusetts, 1974, pg. 264.

(33) Lumry, R., Muller, A., Kokubun, H., Rev. Sci. Instr., 36, 1214 (1965).

(34) Debye, P., Sears, F. W., Proc. Natl. Acad. Sci. U.S., 18, 409 (1932).

(35) Haar, H. P., Hauser, M., Rev. Sci. Instr., 45, 632 (1978).

(36) Maercks, O., Z. Physik., 15, 562 (1936).

(37) Bailey, E. A., Rollefson, G. K., Chem. Phys., 21(8), 1315 (1953).

(38) Birks, J. B., Little, W. A., Proc. Phys. Soc., A66, 921 (1953).

(39) Spencer, R. D., Weber, G., Ann. N. Y. Acad. Sci., 158, 361 (1969).

(40) Mitchell, G. W., Spercer, R. D., Phase-Resolved Nanosecond Spectrofluorometry: Application Brief SLM Instruments, Urbana, Illinois, 1981, 4 p.

(41) Schlag, E. W., Selzle, H. L., Schneider, S., Larsen, J. G., Rev. Sci. Instr., 45, 364 (1974).

(42) Gugger, H., Calzaferri, G., J. Photo., 13, 21 (1979).

(43) Gugger, H., Calzaferri, G., J. Photo., 13, 295 (1980).

(44) Chandler, D. W., Rev. Sci. Instr., 46(6), 786 (1975).

(45) Keyser, L. F., Levine, S. Z., Kaufman, F., J. Chem. Phys., 54, 355 (1971).

(46) Hauser, M., Heidt, G., Rev. Sci. Instr., 46, 470 (1975).

(47) Heidt, G., Appl. Spec., 30, 553 (1976).

(48) Robbins, R. J., Flemming, G. R., Beddard, G. S., Robinson, G. W., Thistlethwaite, P. J., Woolfe, G. J., J. Am. Chem. Soc., 102, 6271 (1980).

(49) Shapiro, S. L., Winn, K. R., J. Chem. Phys., 73, 5958 (1980).

(50) Huder, J. R., J. Opt. Soc., 56, 1195 (1966).

(51) Link, J. K., J. Opt. Soc., 56, 1195 (1966).

(52) Foster, E. W., in Reports on Progress in Physics (A. C. Strickland, ed.), The Institute of Physics and The Physical Society, London, 1963, 512 p.

(53) Schlag, E. W., Weysenhoff, H. V., J. Chem. Phys., 51, 2508 (1969).

(54) Leskovar, B., Lo, C. C., Rev. Sci. Instr., 47, 1113 (1976).

(55) Kremer, J. M. H., Esker, M. W. J., Biochemistry, 16, 3972 (1977).

(56) Berlman, I. B., Handbook of Fluorescence Spectra of Aromatic Molecules, Academic Press, New York, 1971, 466 p.

(57) Maclaren, M. D., Marsaglia, G., J. Ass. Comp. Man., 12, 83 (1965).

(58) Runge, P. K., Optics Communications, 3, 434 (1971).

(59) Runge, P. K., Optics Communications, 4, 195 (1971).

APPENDIX A

BASIC LANGUAGE PHASE ANGLE SHIFT PROGRAMS

```
LIST

1  REM  THE NAME OF THIS IS PLOTD
2  DIM Y2(1024),Y1PL(1024),A$
3 D$ = "": PRINT D$;"NOMON I,O"
4  CALL  - 936
6  PRINT "DO YOU WANT A CONTINUOUS LINE ": INPUT "OR DATA POINTS? (L OR P)
    ";L$
21  CALL  - 936: INPUT "PLOT SIZE IN INCHES? ";YS
22  INPUT "AND VERTICAL OFFSET IN INCHES? ";OS
23  IF OS + YS > 8 THEN 21
24  IF OS < 0 THEN 21
25  IF OS = 0 THEN 28
26 OS = OS * 200: PRINT D$;"PR#2"
27  FOR K = 1 TO OS: PRINT  CHR$ (112): NEXT K
28  PRINT D$;"PR#0": CALL  - 936
32  INPUT "DO YOU WANT LOG DISPLAYED AFTERWARDS? Y OR N ";Z$
34  INPUT "DO YOU WANT LOG ONLY?Y/N ";Q$
40  PRINT "INPUT TOTAL NUMBER OF DATA POINTS": PRINT "YOU WANT PLOTTED...T
    HIS IS FUN"
50  INPUT PT
55  INPUT "ENTER FILE NAME ";A$
56 V$ = "V0"
57 S$ = "S6"
58 X$ = "D2"
60  PRINT D$;"OPEN";A$;",";V$;",";S$;",";X$
62  PRINT D$;"READ";A$: FOR B = 1 TO PT: INPUT Y2(B): NEXT B: PRINT D$;"CL
OSE";A$
64 MX = 0
65  FOR K = 1 TO PT
66  IF MX < Y2(K) THEN 68
67  GOTO 70
68 MX = Y2(K)
70  NEXT K
72 YS = (MX / 63000.0) * 40 * 8 / YS
100  IF Q$ <  > "N" THEN 918
105  FOR J = 1 TO PT
110 Y1PL(J) =  INT (Y2(J) / YS + 0.5)
120  NEXT J
130  IF L$ = "P" THEN 800
150  PRINT D$;"PR#2"
155  PRINT  CHR$ (122), CHR$ (122)
160 XS = 2
165 YC = 0
170  FOR I = 1 TO PT
175 INCRY = Y1PL(I) - YC
180 YC = Y1PL(I)
185  IF INCRY < 0 THEN 235
190  IF INCRY = 0 THEN 225
195  PRINT  CHR$ (114), CHR$ (114)
200  FOR L = 1 TO INCRY
205  PRINT  CHR$ (112)
210  NEXT L
220  NEXT I
221  GOTO 400
225  PRINT  CHR$ (114), CHR$ (114)
230  NEXT I
231  GOTO 400
235  PRINT  CHR$ (114), CHR$ (114)
236 IZ =  ABS (INCRY)
```

```
240  FOR L = 1 TO IZ
245  PRINT  CHR$ (116)
250  NEXT L
256  NEXT I
260  GOTO 400
400 K6 = Y1PL(PT)
405  PRINT D$;"PR#2"
407  PRINT  CHR$ (121), CHR$ (32), CHR$ (32), CHR$ (32)
410  FOR K7 = 1 TO K6
420  PRINT  CHR$ (116)
430  NEXT K7
440  FOR K7 = 1 TO PT
450  PRINT  CHR$ (118), CHR$ (118)
460  NEXT K7
500  GOTO 916
800  PRINT D$;"PR#2"
805  PRINT  CHR$ (122), CHR$ (122), CHR$ (121)
810 XS = 2
815 YC = 0
820  FOR I = 1 TO PT
825 INCRY = Y1PL(I) - YC
830 YC = Y1PL(I)
835  IF INCRY < 0 THEN 885
840  IF INCRY = 0 THEN 875
845  PRINT  CHR$ (114), CHR$ (114)
850  FOR L = 1 TO INCRY
855  PRINT  CHR$ (112)
860  NEXT L
865  PRINT  CHR$ (122), CHR$ (122), CHR$ (121), CHR$ (32)
870  NEXT I
871  GOTO 400
875  PRINT  CHR$ (114), CHR$ (114), CHR$ (122), CHR$ (122), CHR$ (121), CHR$
     (32)
880  NEXT I
881  GOTO 400
885  PRINT  CHR$ (114), CHR$ (114)
886 IZ =  ABS (INCRY)
890  FOR L = 1 TO IZ
895  PRINT  CHR$ (116)
900  NEXT L
905  PRINT  CHR$ (122), CHR$ (122), CHR$ (121), CHR$ (32)
906  NEXT I
910  GOTO 400
913  GOTO 1000
916  IF Z$ = "Y" THEN 918
917  GOTO 980
918 Z$ = "N"
920  FOR J8 = 1 TO PT
925 YZ = Y2(J8): IF YZ <  = 0 THEN YZ = 1
926 Y1PL(J8) =  INT (150 *  LOG (YZ / YS) + 0.5)
927  IF Y1PL(J8) < 0 THEN Y1PL(J8) = 0
930  NEXT J8
950  GOTO 150
980  INPUT "DO YOU WANT ANOTHER FILE PLOTTED?(Y OR N) ";MG$
981  IF MG$ = "Y" THEN 4
1000  END
```

```
PR#0
]LOAD PLOTD
]LIST

1  REM  THE NAME OF THIS IS PLOTD

2  REM  THIS PROGRAM WILL PLOT A
     DATA SET
3  DIM Y2(1024),Y1PL(1024),A$
4 D$ = "": PRINT D$;"NOMON I,O"
5  CALL  - 936
6  PRINT "DO YOU WANT A CONTINUOU
     S LINE ": INPUT "OR DATA POI
     NTS? (L OR P) ";L$
7  REM
8  REM  INPUT DATA SECTION
9  REM
21  CALL  - 936: INPUT "PLOT SIZE
     IN INCHES? ";YS
22  INPUT "AND VERTICAL OFFSET IN
     INCHES? ";OS
23  IF OS + YS > 8 THEN 21
24  IF OS < 0 THEN 21
25  IF OS = 0 THEN 28
26 OS = OS * 200: PRINT D$;"PR#2"
```

```
]?"

?SYNTAX ERROR
]LOAD MASTER
]?""


]LIST

1  REM
2  REM  THIS PROGRAM WILL DIAL UP THE
3  REM  IBM 370/168 MODEL 3 COMPUTER
4  REM
5  REM  INPUT DATA SECTION
6  REM
50  INPUT "DO YOU WANT A PLOT ?(Y/N)";NN$
55  IF NN$ <  > "Y" THEN 100
56  PRINT ""
59  FLASH
60  PRINT "PLEASE PUT PLOTTER IN UPPER LEFT CORNER."
61  NORMAL
90  INPUT "NUMBER OF DATA POINTS ?";NK
100  DIM ITEM$(1024),G(1024),CP$(100)
102 D$ = ""
104  REM :CONTROL-D
106  PRINT D$;"NOMON C,I,O"
108  INPUT "WHAT LAMP FILENAME ?";A$
110  INPUT "WHAT SLOT NUMBER ?";S1$
112  INPUT "WHAT DECAY FILENAME ?";AB$
114  INPUT "WHAT SLOT NUMBER ?";AS1$
116 S1$ = "S" + S1$
118 AS1$ = "S" + AS1$
120 FE$ = A$:FB$ = S1$:FC$ = AB$:FD$ = AS1$
122 NP = NK
124  INPUT "WHAT RESIDUAL FILENAME ?";T$
126  INPUT "WHAT CORRELATION FILENAME ?";TB$
140  INPUT "WHAT SLOT FOR RESIDUALS ?";S$
142  INPUT "WHAT SLOT FOR CORRELATION ?";SB$
144 FT$ = T$:FF$ = TB$:FS$ = "S" + S$:FG$ = "S" + SB$
150 Y$ = T$
152 T$ = T$ + ",S" + S$ + ",V0"
155  INPUT "TITLE:";ZA$
158  INPUT "GUESS A1(F5.2) ";A1$
160  INPUT "A2 ";A2$
162  INPUT "TAU 1 ";T1$
164  INPUT "TAU 2 ";T2$
166  INPUT "CHANNEL FROM ";TT
168  INPUT "TO ";SS
170 TT$ =  STR$ (TT):SS$ =  STR$ (SS)
172  IF TT < 1000 THEN 177
174  GOTO 178
177 TT$ = "0" + TT$
178  IF SS < 1000 THEN 183
180  GOTO 185
183 SS$ = "0" + SS$
185  PRINT ""
190  FLASH
192  PRINT "PLEASE DIAL #822": NORMAL
193  PRINT
```

```
194  FOR I = 1 TO 4000: NEXT I
195  PRINT ""
196  PRINT "SET PHONE ON MODEM"
197  PRINT
198  FLASH : PRINT "CD LIGHT MUST BE ON IN 25 SEC.": NORMAL
199  FOR I = 1 TO 10000: NEXT I: GOSUB 4110
200  REM  LOGON SECTION
201  PRINT D$;"PR#1"
202  PRINT "LOGON U12288A/DOIT NON"
204  PRINT D$;"IN#1"
206  GOSUB 900
215  PRINT "TSOSTRT"
220  GOSUB 1059
230  REM  CLEAR OLD FILE ON THE
231  REM  IBM 370/168
260  PRINT "DELETE LAMP.DATA"
261  GOSUB 990
266  PRINT "DELETE DECAY.DATA"
269  GOSUB 990
272  PRINT "DELETE RES.DATA"
273  GOSUB 990
278  PRINT "DELETE CDATA.DATA"
282  GOSUB 990
286  PRINT "EDIT LAMP DATA NEW"
287  GOSUB 950
288  PRINT D$;"PR#0"
289  PRINT D$;"IN#0"
290  GOSUB 1700
292 X =  FRE (0)
295  GOSUB 500
298  PRINT " END SAVE"
302  GOSUB 950
303  REM  EDIT NEW DATASETS
306  PRINT "EDIT DECAY DATA NEW"
307  GOSUB 950
308  PRINT D$;"PR#0"
310  PRINT D$;"IN#0"
311 S1$ = AS1$:A$ = AB$
312  GOSUB 1700
313 X =  FRE (0)
314  GOSUB 500
320  PRINT " END SAVE"
324  GOSUB 950
325 X =  FRE (0)
326  PRINT "COR"
328  GOSUB 950
330  GOSUB 1329
332 X =  FRE (0)
400  PRINT "EDIT RES DATA OLD"
404  GOSUB 990
408 X =  FRE (0)
412 WR = (NP / 10) + 0.9
414 NK =  INT (WR)
418  GOSUB 700
428  PRINT " END"
430  GOSUB 950
431 X =  FRE (0): PRINT "EDIT CDATA DATA OLD"
432  GOSUB 990
433 Y$ = TB$:T$ = TB$ + ",S" + SB$ + ",VO"
435 NK =  INT (NK / 2)
```

```
436 X =  FRE (0)
438  GOSUB 700
440 X =  FRE (0)
444  PRINT " END"
448  GOSUB 950
450  GOSUB 1000
452  PRINT " LOGOFF"
455  PRINT D$;"PR#0"
456  PRINT D$;"IN#0"
457  GOSUB 6100
458  GOSUB 5000
460  GOTO 2000
465  REM  TRANSFER DATA SECTION
499  REM :CONTROL-A,H,R
500 DD$ = ""
501  PRINT D$;"PR#0"
508  FLASH : PRINT "DATA TRANSFER": NORMAL
510 BIG = 999999
512 C$ = "":DD$ = ""
514  REM :C$ AND DD$ ARE BLANK
516  FOR I = 1 TO NK
518  IF G(I) > BIG THEN 580
520 G(I) = G(I) + 1000000
522  NEXT I
524  PRINT D$;"PR#1"
525  REM  THIS LOOP CHAINS EACH DATA VALUE TOGETHER AS A STRING
526  FOR I = 1 TO 10: IF I > NK THEN 588
528 C$ =  STR$ (G(I)):DD$ = DD$ + C$: NEXT I
529  REM  PRINT THE STRING VALUE
530  PRINT DD$
532  PRINT
534  PRINT D$;"IN#1"
536  FOR I = 1 TO 5000: NEXT I
538 C$ = "":DD$ = ""
540  REM :C$ AND DD$ ARE BLANK
542 I = 1
544 I = I + 10
546 M = I + 9
548  FOR N = I TO M
550  IF N > NK THEN 588
552 C$ =  STR$ (G(N))
554 DD$ = DD$ + C$
556  NEXT N
558  PRINT DD$
560  PRINT D$;"PR#0"
562  GET B$
566  PRINT B$
570  IF B$ >  < "" THEN 562
571  PRINT D$;"PR#1"
572  REM :CONTROL-S
574 C$ = "":DD$ = ""
576  REM :C$ AND DD$ ARE BLANK
578  GOTO 544
580  PRINT D$;"PR#0"
581  REM  ERROR SUBROUTINE FOR INTEGER VALUES GREATER THAN 1 MILLION
582  FLASH
584  PRINT "***ERROR-INPUT GREATER THEN 999,999***"
586  NORMAL : GOTO 7000
588  PRINT DD$
590  PRINT D$;"PR#0"
```

```
592  GET B$
596  PRINT B$
600  IF B$ <  > "" THEN 592
602  REM :CONTROL-S
604  FOR I = 1 TO 7000: NEXT I
605  PRINT D$;"PR#1"
615  RETURN
700  FOR I = 1 TO NK:ITEM$(I) = "0": NEXT I:X =  FRE (0): PRINT "LIST"
705  PRINT D$,"PR#0"
706  PRINT NK
708  INPUT Z$
709  REM  READ DATA SECTION FOR RESIDUAL DATA
710  FOR I = 1 TO NK: INPUT ITEM$(I): NEXT I
715  INPUT Z$
720  FLASH : PRINT "DATA TRANSFER": NORMAL
722 J =  - 9:I = 0
724 J = J + 10:I = I + 1
726  IF I > NK THEN 776
728 A$ = ITEM$(I)
730 B$ =  MID$ (A$,1,6):C$ =  MID$ (A$,8,6):F$ =  MID$ (A$,15,6)
732 P$ =  MID$ (A$,22,6):S$ =  MID$ (A$,29,6):R$ =  MID$ (A$,36,6)
734 U$ =  MID$ (A$,43,6):W$ =  MID$ (A$,50,6):Q$ =  MID$ (A$,57,6):X$ =  MID$
     (A$,64,6)
736  IF J > NP THEN
738 G(J) =  VAL (B$)
740  IF (J + 1) > NP THEN 776
742 G(J + 1) =  VAL (C$)
744  IF (J + 2) > NP THEN 776
746 G(J + 2) =  VAL (F$)
748  IF (J + 3) > NP THEN 776
749 G(J + 3) =  VAL (P$)
750  IF (J + 4) > NP THEN 776
752 G(J + 4) =  VAL (S$)
754  IF (J + 5) > NP THEN 776
756 G(J + 5) =  VAL (R$)
758  IF (J + 6) > NP THEN 776
760 G(J + 6) =  VAL (U$)
762  IF (J + 7) > NP THEN 776
764 G(J + 7) =  VAL (W$)
766  IF (J + 8) > NP THEN 776
768 G(J + 8) =  VAL (Q$)
770  IF (J + 9) > NP THEN 776
772 G(J + 9) =  VAL (X$)
774  GOTO 724
776  PRINT D$;"OPEN";T$
778  PRINT D$;"WRITE";Y$
780  FOR I = 1 TO NP: PRINT G(I): NEXT I
782  PRINT D$;"CLOSE"
784  FOR I = 1 TO NP
788  NEXT I
790  PRINT D$;"PR#1"
792  PRINT D$;"IN#0"
796  RETURN
830  PRINT D$;"PR#0"
832  PRINT D$;"IN#1"
834 EH$ = ""
836  FOR I = 1 TO 11: GET ITEM$(I): NEXT I
837  FOR I = 1 TO 11:EH$ = EH$ + ITEM$(I): NEXT I
839  FOR I = 1 TO 7000: NEXT I
840  PRINT EH$
```

```
436 X =  FRE (0)
438  GOSUB 700
440 X =  FRE (0)
444  PRINT " END"
448  GOSUB 950
450  GOSUB 1000
452  PRINT " LOGOFF"
455  PRINT D$;"PR#0"
456  PRINT D$;"IN#0"
457  GOSUB 6100
458  GOSUB 5000
460  GOTO 2000
465  REM  TRANSFER DATA SECTION
499  REM :CONTROL-A,H,R
500 DD$ = ""
501  PRINT D$;"PR#0"
508  FLASH : PRINT "DATA TRANSFER": NORMAL
510 BIG = 999999
512 C$ = "":DD$ = ""
514  REM :C$ AND DD$ ARE BLANK
516  FOR I = 1 TO NK
518  IF G(I) > BIG THEN 580
520 G(I) = G(I) + 1000000
522  NEXT I
524  PRINT D$;"PR#1"
525  REM  THIS LOOP CHAINS EACH DATA VALUE TOGETHER AS A STRING
526  FOR I = 1 TO 10: IF I > NK THEN 588
528 C$ =  STR$ (G(I)):DD$ = DD$ + C$: NEXT I
529  REM  PRINT THE STRING VALUE
530  PRINT DD$
532  PRINT
534  PRINT D$;"IN#1"
536  FOR I = 1 TO 5000: NEXT I
538 C$ = "":DD$ = ""
540  REM :C$ AND DD$ ARE BLANK
542 I = 1
544 I = I + 10
546 M = I + 9
548  FOR N = I TO M
550  IF N > NK THEN 588
552 C$ =  STR$ (G(N))
554 DD$ = DD$ + C$
556  NEXT N
558  PRINT DD$
560  PRINT D$;"PR#0"
562  GET B$
566  PRINT B$
570  IF B$ > < "" THEN 562
571  PRINT D$;"PR#1"
572  REM :CONTROL-S
574 C$ = "":DD$ = ""
576  REM :C$ AND DD$ ARE BLANK
578  GOTO 544
580  PRINT D$;"PR#0"
581  REM  ERROR SUBROUTINE FOR INTEGER VALUES GREATER THAN 1 MILLION
582  FLASH
584  PRINT "***ERROR-INPUT GREATER THEN 999,999***"
586  NORMAL : GOTO 7000
588  PRINT DD$
590  PRINT D$;"PR#0"
```

```
846  PRINT D$;"PR#1"
848  RETURN
849  REM  RETREIVE DATA SUBROUTINES (FROM CLIST PROGRAMS)
850  PRINT D$;"PR#0"
852  PRINT D$;"IN#1"
854  INPUT BB$
856  INPUT BB$
858  INPUT BB$
860  INPUT BB$
862  INPUT BB$
864  FOR I = 1 TO 2000: NEXT I
866  PRINT D$;"PR#1"
868  RETURN
900  PRINT D$;"PR#0"
901  PRINT D$;"IN#1"
902 EH$ = ""
903  GET BB$:EH$ = EH$ + BB$
906  IF BB$ <  > "" THEN 903
908  REM :CONTROL-S
910  PRINT "S$"
915  PRINT EH$
920  FOR I = 1 TO 7000: NEXT I
922  PRINT D$;"PR#1"
925  RETURN
930  PRINT D$;"PR#0"
932  PRINT D$;"IN#1"
934  INPUT BB$
936  INPUT BB$
938  INPUT BB$
940  INPUT BB$
942  FOR I = 1 TO 2000: NEXT I
944  PRINT D$;"PR#1"
946  RETURN
947  REM  BREAK LINE SUBROUTINE FOR TERMINATION OF LINE FEED
950  PRINT D$;"PR#0"
951  PRINT D$;"IN#1"
952  GET EI$
953  IF EI$ <  > "" THEN 952
954 EH$ = ""
956  PRINT "S$"
958  GET EI$
964 EH$ = EH$ + EI$
965  IF EI$ <  > "" THEN 958
966  PRINT EH$
967  PRINT D$;"PR#1"
970  FOR I = 1 TO 1000: NEXT I
975  RETURN
976  REM  INPUT DATA FROM FORTRAN PROGRAM OKLA1
980  PRINT D$;"PR#0"
981  INPUT BB$
982  INPUT BB$
986  FOR I = 1 TO 2000: NEXT I
987  PRINT D$;"PR#1"
988  RETURN
990  PRINT D$,"PR#0"
991  PRINT D$,"IN#1"
992  INPUT BB$
993  INPUT BB$
994  INPUT BB$
996  FOR I = 1 TO 2000: NEXT I
```

```
997  PRINT D$,"PR#1"
998  RETURN
1000 X =  FRE (0)
1001  PRINT "TIME"
1003  GOSUB 830
1008 TI$ = EH$
1010  PRINT "DATE"
1012  GOSUB 980
1024 DA$ = BB$
1030  PRINT D$;"PR#1"
1040  PRINT D$;"IN#0"
1045  FOR I = 1 TO 4000: NEXT I
1050  RETURN
1059  PRINT D$,"PR#0"
1060  PRINT D$,"IN#1"
1061  INPUT ZZ$
1062  INPUT ZZ$
1063  INPUT ZZ$
1064  INPUT ZZ$
1065  INPUT ZZ$
1066  INPUT ZZ$
1067  INPUT ZZ$
1068  INPUT ZZ$
1069  INPUT ZZ$
1070  INPUT ZZ$
1071  INPUT ZZ$
1072  INPUT ZZ$
1073  INPUT ZZ$
1074  INPUT ZZ$
1075  INPUT ZZ$
1076  PRINT D$,"PR#1"
1077  PRINT D$,"IN#0"
1078  FOR I = 1 TO 2000: NEXT I
1080  RETURN
1272  INPUT ZZ$
1329 NK$ =  STR$ (NK)
1330  IF NK < 1000 THEN 1334
1332  GOTO 1338
1334 NK$ = "0" + NK$
1338  PRINT NK$
1346  GOSUB 930
1352  PRINT "1"
1354  GOSUB 980
1355  PRINT ZA$
1356  GOSUB 990
1357  PRINT "0.0"
1358  GOSUB 980
1360  PRINT "0.0"
1362  GOSUB 850
1370  PRINT "1"
1374  GOSUB 990
1380  PRINT TT$
1381  FOR I = 1 TO 4000: NEXT I
1382  PRINT SS$
1384  GOSUB 930
1390  PRINT "1"
1392  GOSUB 980
1394  PRINT A1$
1396  GOSUB 980
1398  PRINT A2$
```

```
1400  GOSUB 980
1402  PRINT T1$
1403  GOSUB 980
1404  PRINT T2$
1405  GOSUB 850
1412  PRINT "1"
1413  PRINT D$;"PR#0"
1414  INPUT ZZ$
1416  INPUT KA$
1422  INPUT KB$
1424  INPUT KC$
1426  INPUT KD$
1428  INPUT KE$
1430  INPUT KF$
1432  INPUT KG$
1434  INPUT KH$
1436  INPUT KK$
1438  INPUT ZZ$
1440  INPUT ZZ$
1441  PRINT D$;"PR#1"
1442  FOR I = 1 TO 3000: NEXT I
1443  PRINT "1"
1444  GOSUB 980
1450  RETURN
1451  REM
1452  REM  DELAYED PRINT OUT OF RELEVANT
1453  REM  DATA FROM ALL FORTRAN PROGRAMS
1550  PRINT D$;"PR#0"

1551  PRINT D$;"IN#0"
1552  PRINT KA$
1554  PRINT KB$
1556  PRINT KC$
1558  PRINT KD$
1560  PRINT KE$
1562  PRINT KF$
1564  PRINT KG$
1566  PRINT KH$
1568  PRINT KK$
1570  PRINT
1572  PRINT TI$
1574  PRINT DA$
1600  PRINT D$;"PR#0"
1602  PRINT ""
1603  REM :BELL
1604  PRINT D$;"PR#1"
1606  FOR I = 1 TO 2000: NEXT I
1608  GOTO 7000
1700  PRINT D$;"OPEN";A$;",V0,";S1$
1702  PRINT D$;"READ";A$
1704  FOR I = 1 TO NK: INPUT G(I): NEXT I
1706  PRINT D$;"CLOSE"
1722  RETURN
1800  REM  HARD COPY PLOTTING SUBROUTINE
2000  PRINT ""
2006  PRINT D$,"MON C": PRINT D$,"NOMON I,O"
2007 L$ = "L":OS = 0:Z$ = "N":Q$ = "N"
2008 A$ = FE$:S$ = FB$:NMN = 0
2010  REM  CHAINING OF PRINT COMMANDS FOR RAPID PEN MOVEMENTS
2012 PT = NP
```

```
2013  PRINT D$;"PR#2"
2015 C1$ =  CHR$ (114) +  CHR$ (114)
2016 C2$ =  CHR$ (112)
2017 C3$ =  CHR$ (116)
2018 C4$ =  CHR$ (118) +  CHR$ (118)
2019 C5$ = C4$ + C4$ + C4$ + C4$ + C4$ + C4$ + C4$ + C4$ + C4$ + C4$
2028  PRINT D$;"PR#0": CALL  - 936
2059  REM  READ RESIDUAL DATA FOR PLOT
2060  PRINT D$;"OPEN";A$;",V0";",";S$
2062  PRINT D$;"READ";A$: FOR B = 1 TO PT: INPUT G(B): NEXT B: PRINT D$;"C
     LOSE";A$
2064 MX = 0:NMN = NMN + 1:ML = 0
2065  FOR K = 1 TO PT
2066  IF ML > G(K) THEN ML = G(K)
2067  NEXT K
2068  FOR K = 1 TO PT
2069  IF MX < G(K) THEN MX = G(K)
2070  NEXT K
2072 YS = (MX / 63000.0) * 40 * 8 / YS
2100  IF Q$ <  > "N" THEN 2918
2105  FOR J = 1 TO PT
2110 ITEM$(J) =  STR$ ( INT (G(J) / YS + 0.5))
2120  NEXT J
2130  IF L$ = "P" THEN 2800
2150  PRINT D$;"PR#2"
2155  PRINT  CHR$ (122), CHR$ (122)
2160 XS = 2
2165 YC = 0
2170  FOR I = 1 TO PT
2175 INCRY =  VAL (ITEM$(I)) - YC
2180 YC =  VAL (ITEM$(I))
2185  IF INCRY < 0 THEN 2235
2190  IF INCRY = 0 THEN 2225
2195  PRINT C1$
2200  FOR L = 1 TO INCRY
2205  PRINT C2$
2210  NEXT L
2220  NEXT I
2221  GOTO 2400
2225  PRINT C1$
2230  NEXT I
2231  GOTO 2400
2235  PRINT C1$
2236 IZ =  ABS (INCRY)
2240  FOR L = 1 TO IZ
2245  PRINT C3$
2250  NEXT L
2256  NEXT I
2260  GOTO 2400
2400 K6 =  VAL (ITEM$(PT))
2405  PRINT D$;"PR#2"
2407  PRINT  CHR$ (121), CHR$ (32), CHR$ (32), CHR$ (32)
2410  FOR K7 = 1 TO K6
2420  PRINT  CHR$ (116)
2430  NEXT K7
2440 PB = PT / 50
2441  FOR K7 = 1 TO PB
2450  PRINT C5$;C5$;C5$;C5$C5$
2460  NEXT K7
2500  GOTO 2916
```

```
2800  PRINT D$;"PR#2"
2805  PRINT  CHR$ (122), CHR$ (122), CHR$ (121)
2810 XS = 2
2815 YC = 0
2820  FOR I = 1 TO PT
2825 INCRY =  VAL (ITEM$(I)) - YC
2830 YC =  VAL (ITEM$(I))
2835  IF INCRY < 0 THEN 2885
2840  IF INCRY = 0 THEN 2875
2845  PRINT  CHR$ (114), CHR$ (114)
2850  FOR L = 1 TO INCRY
2855  PRINT  CHR$ (112)
2860  NEXT L
2865  PRINT  CHR$ (122), CHR$ (122), CHR$ (121), CHR$ (32)
2870  NEXT I
2871  GOTO 2400
2875  PRINT  CHR$ (114), CHR$ (114), CHR$ (122), CHR$ (122), CHR$ (121), CHR$
     (32)
2880  NEXT I
2881  GOTO 2400
2885  PRINT  CHR$ (114), CHR$ (114)
2886 IZ =  ABS (INCRY)
2890  FOR I = 1 TO IZ
2895  PRINT  CHR$ (116)
2900  NEXT L
2905  PRINT  CHR$ (122), CHR$ (122), CHR$ (121), CHR$ (32)
2906  NEXT I
2910  GOTO 2400
2913  GOTO 3000
2916  IF Z$ = "Y" THEN 2918
2917  GOTO 2960
2918 Z$ = "N"
2920  FOR J8 = 1 TO PT
2925 YZ = G(J8): IF YZ <  = 0 THEN YZ = 1
2926 ITEM$(J8) =  STR$ ( INT (200 *  LOG (YZ / YS) + 0.5))
2927  IF  VAL (ITEM$(J8) < 0 THEN Y1PLK(J8) = 0
2930  NEXT J8
2950  GOTO 2150
2960 KJK = KJK + 1
2970  IF  ABS (ML) > MX THEN KA$(NMN) =  STR$ (ML)
2972  IF MX >  ABS (ML) THEN KA$(NMN) =  STR$ (MX)
2981  IF HJH = 1 THEN 3130
2982  IF LLK = 1 THEN 3100
2983 A$ = FC$:S$ = FD$
2985 YS = 2
2990  IF KJK < 2 THEN 2015
3000  PRINT  CHR$ (122)
3003  FOR I = 1 TO 400
3005  PRINT  CHR$ (114); CHR$ (114); CHR$ (114); CHR$ (114); CHR$ (114)
3010  NEXT I
3015  PRINT  CHR$ (121)
3020  FOR I = 1 TO 20
3022  PRINT C5$;C5$;C5$;C5$;C5$
3025  NEXT I
3026  GOSUB 6490
3027  PRINT  CHR$ (122)
3030  FOR I = 1 TO 50
3035  PRINT  CHR$ (112); CHR$ (112); CHR$ (112); CHR$ (112); CHR$ (112)
3040  NEXT I
3050  FOR I = 1 TO 100
```

```
3055  PRINT  CHR$ (116); CHR$ (116); CHR$ (116); CHR$ (116); CHR$ (116)
3060  NEXT I
3061  FOR I = 1 TO 400: PRINT  CHR$ (114); CHR$ (114); CHR$ (114); CHR$ (1
     14); CHR$ (114): NEXT I
3062  PRINT  CHR$ (121): FOR I = 1 TO 20: PRINT C5$;C5$;C5$;C5$;C5$: NEXT
     I
3070 YS = 0.25
3075 A$ = FT$:S$ = FS$
3080 LLK = LLK + 1
3090  IF LLK = 1 THEN 2015
3100  REM :PRINT
3104  PRINT  CHR$ (121)
3105  FOR I = 1 TO 135
3110  PRINT  CHR$ (113); CHR$ (113); CHR$ (113); CHR$ (113); CHR$ (113): NEXT
     I
3111  FOR I = 1 TO 325: PRINT  CHR$ (114): NEXT I
3112  PRINT  CHR$ (122): FOR I = 1 TO 200: PRINT  CHR$ (114); CHR$ (114); CHR$
     (114); CHR$ (114); CHR$ (114): NEXT I
3113  PRINT  CHR$ (121): FOR I = 1 TO 200: PRINT  CHR$ (118); CHR$ (118); CHR$
     (118); CHR$ (118); CHR$ (118): NEXT I
3116 A$ = FF$:S$ = FG$
3117 YS = 0.5:PT =  INT (NP / 2)
3118 HJH = HJH + 1
3120  IF HJH = 1 THEN 2015
3130 INCRY = 2 * INCRY: IF INCRY > 0 THEN  GOTO 3134
3131  IF INCRY = 0 THEN  GOTO 3135
3132 INCRY =  ABS (INCRY): FOR I = 1 TO INCRY: PRINT  CHR$ (116): NEXT I
3133  GOTO 3135
3134  FOR I = 1 TO INCRY: PRINT  CHR$ (112): NEXT I
3135  GOSUB 6300: GOTO 1550
4110 Q$ = "NUM"
4130  PRINT D$;"NOMON I,O"
4154  PRINT D$;"OPEN";Q$
4156  PRINT D$;"READ";Q$
4158  FOR J = 1 TO 100: INPUT CP$(J): NEXT J
4159  PRINT D$;"CLOSE";Q$
4160  RETURN
4492  FOR V = 1 TO  LEN (B$):C$ =  MID$ (B$,V,1)
4494 Y =  ASC (C$)
4496  PRINT CP$(Y)
4497  NEXT V
4500  RETURN
4550  REM  THESE SUBROUTINES SET UP THE PLOTTING SCALES
5000  IF NN$ <  > "Y" THEN 1550
5001  PRINT D$;"PR#2": FOR I = 1 TO 425: PRINT  CHR$ (112): NEXT I
5002  FOR I = 1 TO 100: PRINT  CHR$ (114): NEXT I
5003 B$ = KE$
5004  GOSUB 6000
5008  FOR I = 1 TO 1526: PRINT  CHR$ (118): NEXT I
5036 B$ = KF$
5038  GOSUB 6000
5040  FOR I = 1 TO 1270: PRINT  CHR$ (118): NEXT I
5042 B$ = KG$
5044  GOSUB 6000
5046  FOR I = 1 TO 1270: PRINT  CHR$ (118): NEXT I
5048 B$ = KH$
5050  GOSUB 6000
5052  FOR I = 1 TO 1270: PRINT  CHR$ (118): NEXT I
5054 B$ = KK$
5056  GOSUB 6000
```

```
5058  FOR I = 1 TO 1270: PRINT  CHR$ (118): NEXT I
5062 B$ = TI$
5064  GOSUB 4492
5066  FOR I = 1 TO 50: PRINT  CHR$ (114): NEXT I
5068 B$ = DA$
5070  GOSUB 6000
5072  FOR I = 1 TO 860: PRINT  CHR$ (112): NEXT I
5073  FOR I = 1 TO 650: PRINT  CHR$ (118): NEXT I
5076  RETURN
6000  GOSUB 4492
6002  FOR I = 1 TO 60: PRINT  CHR$ (116): NEXT I
6004  RETURN
6100  PRINT D$;"PR#2"
6110  PRINT  CHR$ (121)
6115 C$ =  CHR$ (112) +  CHR$ (112) +  CHR$ (112) +  CHR$ (112) +  CHR$ (1
     12) +  CHR$ (112) +  CHR$ (112) +  CHR$ (112) +  CHR$ (112) +  CHR$ (
     112)
6117 DD$ =  CHR$ (114) +  CHR$ (114) +  CHR$ (114) +  CHR$ (114) +  CHR$ (
     114) +  CHR$ (114) +  CHR$ (114) +  CHR$ (114) +  CHR$ (114) +  CHR$
     (114)
6118 BB$ =  CHR$ (116) +  CHR$ (116) +  CHR$ (116) +  CHR$ (116) +  CHR$ (
     116) +  CHR$ (116) +  CHR$ (116) +  CHR$ (116) +  CHR$ (116) +  CHR$
     (116)
6119 EH$ =  CHR$ (118) +  CHR$ (118) +  CHR$ (118) +  CHR$ (118) +  CHR$ (
     118) +  CHR$ (118) +  CHR$ (118) +  CHR$ (118) +  CHR$ (118) +  CHR$
     (118)
6120  FOR I = 1 TO 65: PRINT  CHR$ (114): NEXT I
6130  FOR I = 1 TO 100: PRINT  CHR$ (112): NEXT I
6140  PRINT  CHR$ (122)
6150  FOR I = 1 TO 40: PRINT C$: NEXT I
6160  FOR I = 1 TO 170: PRINT DD$: NEXT I
6170  FOR I = 1 TO 40: PRINT BB$: NEXT I
6180  FOR I = 1 TO 170: PRINT EH$: NEXT I
6185  PRINT  CHR$ (121)
6190  FOR I = 1 TO 100: PRINT  CHR$ (116): NEXT I
6200  FOR I = 1 TO 65: PRINT  CHR$ (118): NEXT I
6220  RETURN
6300 ITEM$(20) = "CFL " + " MAX=" + KA$(4)
6302 ITEM$(21) = "LFL-" + FE$ + " MAX=" + KA$(1)
6304 ITEM$(22) = "SFL-" + FC$ + " MAX=" + KA$(2)
6306 ITEM$(23) = "RFL " + " MAX=" + KA$(3)
6350  FOR I = 1 TO 1000: PRINT  CHR$ (118): NEXT I
6400  FOR V = 1 TO  LEN (ZA$):C$ =  MID$ (ZA$,V,1)
6402 Y =  ASC (C$)
6404  PRINT CP$(Y)
6406  NEXT V
6408  FOR V = 1 TO  LEN (ZA$): PRINT EH$;EH$: NEXT V
6410  FOR I = 1 TO 1320: PRINT  CHR$ (114): NEXT I
6412  FOR I = 1 TO 180: PRINT  CHR$ (116): NEXT I
6414  FOR V = 1 TO  LEN (ITEM$(20)):C$ =  MID$ (ITEM$(20),V,1)
6416 Y =  ASC (C$)
6418  PRINT CP$(Y)
6420  NEXT V
6421  PRINT  CHR$ (121)
6422  FOR V = 1 TO  LEN (ITEM$(20)): PRINT EH$;EH$: NEXT V
6423  FOR I = 1 TO 137: PRINT  CHR$ (118): NEXT I
6424  FOR I = 1 TO 50: PRINT  CHR$ (116): NEXT I
6426  FOR V = 1 TO  LEN (ITEM$(21)):C$ =  MID$ (ITEM$(21),V,1)
6428 Y =  ASC (C$)
6430  PRINT CP$(Y)
```

```
6432  NEXT V
6433  FOR V = 1 TO  LEN (ITEM$(21)): PRINT EH$;EH$: NEXT V: FOR I = 1 TO 1
     37: PRINT  CHR$ (118): NEXT I
6434  FOR I = 1 TO 50: PRINT  CHR$ (116): NEXT I
6436  FOR V = 1 TO  LEN (ITEM$(22)):C$ =  MID$ (ITEM$(22),V,1)
6438 Y =  ASC (C$)
6440  PRINT CP$(Y)
6442  NEXT V
6444  FOR I = 1 TO 300: PRINT  CHR$ (116): NEXT I
6445  FOR V = 1 TO  LEN (ITEM$(22)): PRINT EH$;EH$: NEXT V: FOR I = 1 TO 1
     37: PRINT  CHR$ (118): NEXT I
6446  FOR V = 1 TO  LEN (ITEM$(23)):C$ =  MID$ (ITEM$(23),V,1)
6448 Y =  ASC (C$)
6450  PRINT CP$(Y)
6452  NEXT V
6460  RETURN
6490  PRINT  CHR$ (121)
6500  FOR I = 1 TO 200: PRINT  CHR$ (114): NEXT I
6504  FOR I = 1 TO 9
6505  PRINT  CHR$ (122)
6506  FOR J = 1 TO 8: PRINT  CHR$ (112): NEXT J
6508  FOR J = 1 TO 16: PRINT  CHR$ (116): NEXT J
6510  FOR J = 1 TO 8: PRINT  CHR$ (112): NEXT J
6512  PRINT  CHR$ (121)
6514  FOR J = 1 TO 200: PRINT  CHR$ (114): NEXT J
6516  NEXT I
6518  FOR I = 1 TO 20: PRINT C5$;C5$;C5$;C5$;C5$: NEXT I
6520  FOR I = 1 TO 50: PRINT  CHR$ (116): NEXT I
6522  FOR I = 1 TO 180: PRINT  CHR$ (114): NEXT I
6524  FOR I = 2 TO 10:I$ =  STR$ (I - 1):MM$ = "0"
6526 Y =  ASC (I$): PRINT CP$(Y):Y =  ASC (MM$): PRINT CP$(Y)
6528  FOR J = 1 TO 152: PRINT  CHR$ (114): NEXT J
6530  NEXT I
6531  GOSUB 6700
6532  FOR I = 1 TO 50: PRINT  CHR$ (112): NEXT I
6534  FOR I = 1 TO 20: PRINT C5$;C5$;C5$;C5$;C5$: NEXT I
6536  RETURN
6700  FOR I = 1 TO 92: PRINT  CHR$ (118): NEXT I
6701 KS$ = "NSEC"
6702  FOR V = 1 TO  LEN (KS$):C$ =  MID$ (KS$,V,1)
6704 Y =  ASC (C$)
6706  PRINT CP$(Y)
6708  NEXT V
6710  RETURN
7000  END

]
```

```
]?""

]LOAD RES
]LIST

1  REM
2  REM  THIS PROGRAM WILL RECEIVE
3  REM  DATA FILES FROM THE TSO AND
4  REM  STORE THE DATA ON AN APPLE DISKETTE
5  DIM ITEM$(1024),G(1024)
6  REM
7  REM  INPUT DATA SECTION
8  REM
10 D$ = ""
20  REM :CONTROL-D
30  PRINT D$,"NOMON I,O"
40  INPUT "ENTER FILE NAME ";T$
51  INPUT "NO OF POINTS ?";NP
55 WR = (NP / 10) + 0.9
56 NK =  INT (WR)
70  INPUT "WHAT DRIVE FOR DATA ?";N$
75 Y$ = T$
80 T$ = T$ + ",V0,S6,D" + N$
96  REM :CONTL-A,H,R
97  PRINT D$,"PR#1
98  PRINT ""
99  PRINT D$,"PR#0"
100  PRINT D$,"IN#1"
114  INPUT Z$
115  REM  INPUT OF DATA SUBROUTINE FROM
116  REM  THE TSO LINES
120  FOR I = 1 TO NK
130  INPUT ITEM$(I)
140  NEXT I
145  INPUT Z$
150  FLASH : PRINT "DATA TRANSFER": NORMAL
170  REM  RECONFIGURATION DATA TO STORE
171  REM  ON THE APPLE
184 J =  - 9:I = 0
185 J = J + 10:I = I + 1
186  IF I > NK THEN 200
187 A$ = ITEM$(I)
188 B$ =  MID$ (A$,2,6):C$ =  MID$ (A$,9,6):F$ =  MID$ (A$,16,6)
189 P$ =  MID$ (A$,23,6):S$ =  MID$ (A$,30,6):R$ =  MID$ (A$,37,6)
190 U$ =  MID$ (A$,44,6):W$ =  MID$ (A$,51,6):Q$ =  MID$ (A$,58,6):X$ =  MID$
     (A$,65,6)
191  IF J > NP THEN 221
192 G(J) =  VAL (B$)
193  IF (J + 1) > NP THEN 221
194 G(J + 1) =  VAL (C$)
195  IF (J + 2) > NP THEN 221
196 G(J + 2) =  VAL (F$)
197  IF (J + 3) > NP THEN 221
198 G(J + 3) =  VAL (P$)
199  IF (J + 4) > NP THEN 221
200 G(J + 4) =  VAL (S$)
201  IF (J + 5) > NP THEN 221
202 G(J + 5) =  VAL (R$)
```

```
203  IF (J + 6) > NP THEN 221
204 G(J + 6) =  VAL (U$)
205  IF (J + 7) > NP THEN 221
206 G(J + 7) =  VAL (W$)
207  IF (J + 8) > NP THEN 221
209 G(J + 8) =  VAL (Q$)
210  IF (J + 9) > NP THEN 221
211 G(J + 9) =  VAL (X$)
212  GOTO 185
215  REM  WRITE DATA TO A DISK FILE
221  PRINT D$;"OPEN";T$
223  PRINT D$;"WRITE";Y$
225  FOR I = 1 TO NP: PRINT G(I): NEXT I
230  PRINT D$;"CLOSE"
235  REM  PRINT DATA ON THE SCREEN
250  FOR I = 1 TO NP
260  PRINT I,G(I)
270  NEXT I
300  PRINT "TO BREAK TYPE :"
310  FLASH : PRINT "CONTL-A CONTL-S RETURN ": NORMAL

]PR#0
```

```
]?""

]LOAD REC
]LIST

1  REM
2  REM  THOS PROGRAM WILL RECEIVE
3  REM  DATA FILES FROM THE TSO
5  DIM ITEM$(1024),G(1024)
10 D$ = ""
20  REM :CONTROL-D
30  PRINT D$,"NOMON I,O"
40  INPUT "ENTER FILE NAME ";T$
51  INPUT "NO OF POINTS ?";NP
55 WR = (NP / 10) + 0.9
56 NK =  INT (WR)
70  INPUT "WHAT DRIVE FOR DATA ?";N$
75 Y$ = T$
80 T$ = T$ + ",V0,S6,D" + N$
96  REM :CONTL-A,H,R
97  PRINT D$,"PR#1
98  PRINT ""
99  PRINT D$,"PR#0"
100  PRINT D$,"IN#1"
114  INPUT Z$
116  REM  INPUT DATA FROM THE IBM 370/168
120  FOR I = 1 TO NK
130  INPUT ITEM$(I)
132  INPUT M$
134 ITEM$(I) = ITEM$(I) + M$
140  NEXT I
147  INPUT Z$
148  REM  TRANSFER DATA FROM STRING MODE
149  REM  TO INTEGER MODE
150  FLASH : PRINT "DATA TRANSFER": NORMAL
184 J =  - 9:I = 0
185 J = J + 10:I = I + 1
186  IF I > NK THEN 200
187 A$ = ITEM$(I)
188 B$ =  MID$ (A$,8,6):C$ =  MID$ (A$,15,6):F$ =  MID$ (A$,22,6)
189 P$ =  MID$ (A$,29,6):S$ =  MID$ (A$,36,6):R$ =  MID$ (A$,43,6)
190 U$ =  MID$ (A$,50,6):W$ =  MID$ (A$,57,6):Q$ =  MID$ (A$,64,6):X$ =  MID$
     (A$,71,6)
191  IF J > NP THEN 221
192 G(J) =  VAL (B$)
193  IF (J + 1) > NP THEN 221
194 G(J + 1) =  VAL (C$)
195  IF (J + 2) > NP THEN 221
196 G(J + 2) =  VAL (F$)
197  IF (J + 3) > NP THEN 221
198 G(J + 3) =  VAL (P$)
199  IF (J + 4) > NP THEN 221
200 G(J + 4) =  VAL (S$)
201  IF (J + 5) > NP THEN 221
202 G(J + 5) =  VAL (R$)
203  IF (J + 6) > NP THEN 221
204 G(J + 6) =  VAL (U$)
205  IF (J + 7) > NP THEN 221
```

```
206 G(J + 7) =  VAL (W$)
207  IF (J + 8) > NP THEN 221
209 G(J + 8) =  VAL (Q$)
210  IF (J + 9) > NP THEN 221
211 G(J + 9) =  VAL (X$)
212  GOTO 185
215  REM  WRITE OUT DATA FILE
221  PRINT D$;"OPEN";T$
223  PRINT D$;"WRITE";Y$
225  FOR I = 1 TO NP: PRINT G(I): NEXT I
230  PRINT D$;"CLOSE"
250  FOR I = 1 TO NP
260  PRINT I,G(I)
270  NEXT I
300  PRINT "TO BREAK TYPE :"
310  FLASH : PRINT "CONTL-A CONTL-S RETURN ": NORMAL
320  END

]PR#0
```

```
]?""

]LOAD TLIST
]LIST

2  HOME
4  PRINT "***************"
6  PRINT "THIS PROGRAM WILL"
8  PRINT "LIST A DATA FILE."
9  PRINT "***************"
10  DIM ZZ(2000)
11 D$ = ""
20  PRINT D$,"NOMON I,O"
25  PRINT
30  INPUT "ENTER FILE NAME ";A$
35  INPUT "LENGTH OF FILE ? ";X
37  INPUT "WHAT DRIVE NO. ?";N$
38 N$ = ",D" + N$
39 S$ = "S6"
41  REM
42  REM  READ DATA FILE
43  REM
45  PRINT D$,"OPEN";A$;",V0,";S$;N$
50  PRINT D$,"READ";A$
60  FOR I = 1 TO X: INPUT ZZ(I): NEXT I
70  PRINT D$,"CLOSE",A$
80  INPUT "DISPLAYED IN BLOCKS ?";A$:A$ =  LEFT$ (A$,1)
100  IF A$ = "Y" THEN 121
110  FOR I = 1 TO X: PRINT I,ZZ(I): NEXT I
115  GOTO 230
116  REM
117  REM  BLOCK DATA VALUES IN PAGES
118  REM  FOR VIEWING ON THE SCREEN
119  REM
121  PRINT " # OF PTS. PER BLOCK ?"
122  INPUT "(20 FIT ON SCREEN)";N
123  FLASH : PRINT "TO BREAK TYPE B": NORMAL
125 R =  INT (X / N) + 1
130  FOR I = 1 TO R
150  FOR J = (I - 1) * N + 1 TO I * N
160  PRINT J,ZZ(J)
170  NEXT J
180  PRINT "RETURN FOR NEXT BLOCK       ";"PAGE ";I;" OF ";R
190  GET A$
191  HOME
195  IF A$ = "B" THEN 220
200  NEXT I
205  GOTO 230
220  PRINT "BREAK TO END"
230  INPUT "DO YOU WANT TO LIST AGAIN ? ";LS$:LS$ =  LEFT$ (LS$,1)
240  IF LS$ = "Y" THEN 80
250  PRINT D$,"MON I,O"

]
```

```
]?""

]LOAD TAKED
]LIST

5  HOME
6  PRINT "******************"
7  PRINT "TAKE DATA PROGRAM"
8  PRINT "******************"
10  DIM Y1(1024),Y2(1024)
11 D$ = ""
15  PRINT D$,"NOMON I,O"
18  PRINT
20  PRINT "    *** STOP ***"
25  PRINT "INTERRUPT MCA CLOCK"
30  PRINT "    *** **** ***"
35  PRINT
36  REM
37  REM   SET VALUE TO TEST FOR LAST CHANNEL
38  REM
50 T = 0.0001
51  REM
52  REM  INPUT DATA FILE NAMES
53  REM
55  INPUT "ENTER FILE NAME ";A$
56  INPUT "INPUT # OF PTS. ";PT
57  INPUT "DRIVE NUMBER ";N$
58 N$ = ",D" + N$
59 S$ = "S6"
60  INPUT "DATA DISPLAY ? (Y/N)";RR$
61  REM  INPUT DATA FROM THE BUFFER
63 RR$ =  LEFT$ (RR$,1)
65  FOR J = 1 TO 1024
70 X =  PEEK ( - 16320)
80 YA =  PEEK ( - 16188)
90 YB =  PEEK ( - 15360)
100 Y1(J) = YA + YB * 256
120  NEXT J
123  REM  DETERMINE LAST CHANNEL AND
124  REM  ARRANGE DATA POINTS
130  FOR J = 1 TO 1023
140 YA = (Y1(J + 1) + 1) / (Y1(J) + 1)
150  IF YA < T THEN 170
160  NEXT J
170 R = J + 4
180  FOR J = R TO 1024
190 Y2(J - R + 1) = Y1(J)
200  NEXT J
210  FOR J = 1 TO R - 1
220 Y2(1024 - R + 1 + J) = Y1(J)
230  NEXT J
249  REM
250  REM  WRITE DATA TO DISK
251  REM
905  PRINT
910  PRINT "IF THRU TURN OFF NITROGEN"
911  PRINT D$;"OPEN";A$;",V0,";S$;N$
912  PRINT D$;"WRITE";A$: FOR A = 1 TO PT: PRINT Y2(A): NEXT A: PRINT D$;"
```

```
]?""

]LOAD TRANS
]LIST

10 D$ = ""
11  PRINT D$,"NOMON I,O"
20  DIM ZZ( 2000 )
21  REM
22  PRINT
25  PRINT "THIS PRGM. TRANSFERS TEXT FILES"
26  PRINT "FROM DISK TO DISK": PRINT
27  REM
29 S$ = "S6"
30  INPUT "FILE NAME ? ";A$
31  INPUT "LENGTH OF FILE? ";L
32  INPUT "DRIVE TRANSFERING FROM ?";E$
33 N$ = "1": IF E$ = "1" THEN N$ = "2"
34 E$ = ",D" + E$
35 N$ = ",D" + N$
37  REM  INPUT DATA FROM DISK
40  PRINT D$,"OPEN";A$;",VO,";S$;E$
50  PRINT D$,"READ";A$
60  FOR I = 1 TO L: INPUT ZZ(I): NEXT I
65  REM  OUTPUT DATA FROM THE DISK
70  PRINT D$,"CLOSE";A$
80  PRINT D$,"OPEN";A$;",VO,";S$;N$
90  PRINT D$,"WRITE";A$
100  FOR I = 1 TO L: PRINT ZZ( I ): NEXT I
110  PRINT D$,"CLOSE";A$
115  PRINT D$,"MON I,O"
120  END

]
```

```
     CLOSE";A$
915  IF RR$ = "N" THEN 990
916  REM
917  REM  SCALE DATA POINTS FOR GRAPH
918  REM
920  FOR J = 1 TO PT
928  IF MY > Y2(J) THEN 930
929 MY = Y2(J)
930  NEXT J
931 MX = PT * 0.097
935  FOR K = 1 TO PT
945 Y2(K) = Y2(K) / MY * 157
955  IF Y2(K) < 0 THEN Y2(K) = 0
956 Y2(K) =  INT (Y2(K))
959  NEXT K
960 S = 0.1
961  REM
962  REM  PLOT DATA ON SCREEN
963  REM
965  HGR2
966  HPLOT 0,0 TO 0,158 TO 256,158 TO 256,0 TO 0,0
970  FOR I = 1 TO PT
971 AA = AA + 0.097
972 X =  INT ((AA / MX) * 256)
975  HPLOT X,158 - Y2(I)
976  IF X < S * 256 THEN 980
977  HPLOT X,0.0 TO X,10
978 S = S + 0.1
980  NEXT I
990  PRINT D$,"MON I,O"
995  PR# 0
996  PRINT ""
1000  GET Z$: TEXT : HOME : END

]PR#0
```

```
]?""

]LOAD PPD
]LIST

0  HOME
1  PRINT "************************"
2  PRINT
3  PRINT "  PHASE PLANE METHOD "
4  PRINT
5  PRINT "************************"
6  PRINT
7  PRINT
10  DIM F%(1000),D%(1000),W(1000),Z(1000)
15  REM
16  REM  INITIALIZE VARIABLES
17  REM
20  INPUT "# OF DATA PTS. ";NNN
30  IF NNN > 1000 THEN 620
40 TSPAN = 0.097
60  REM :INITIALIZE VARIABLES
70 T2 = 1.96
130  INPUT "WHAT LAMP FILENAME ? ";A$
132  INPUT "WHAT DATA FILENAME ? ";C$
142  INPUT "LAMP AND DATA FILES ARE ON DRIVE ...";S$
144 S$ = "S6,D" + S$
150 D$ = ""
151  REM
152  REM  READ DATA FILES
153  REM
155  PRINT D$,"NOMON C,I,O"
160  PRINT D$,"OPEN";A$;",VO";"," + S$
170  PRINT D$,"READ";A$
180  FOR I = 1 TO NNN: INPUT F%(I): NEXT I
190  PRINT D$,"CLOSE"
220  PRINT D$,"OPEN";C$;",VO";"," + S$
230  PRINT D$,"READ";C$
235  FOR J = 1 TO NNN: INPUT D%(J): NEXT J
236  PRINT D$;"CLOSE"
239  REM
240  REM  SCALE DATA FILES BEFORE CALCULATION
241  REM
242  FOR K = 1 TO NNN
243  IF ML > F%(K) THEN 245
244 ML = F%(K)
245  IF MD > D%(K) THEN 247
246 MD = D%(K):LJ = K
247  NEXT K
248  FOR K = 1 TO NNN
249 KG = D%(K) * ML / MD + 0.5:D%(K) =  INT (KG): NEXT K
251  PRINT
252  PRINT " MATRIX ELEMENT "
253  PRINT " CALCULATION "
254  PRINT
260 NW = NNN - 1
278 K =  - 1
280  FOR L = 1 TO NW
290 K = K + 1
```

```
300 Q = F%(L) + F%(K) + Q
310 R = D%(L) + D%(K) + R
320  IF Q <  = 0 THEN Q = 1
350 W(L) = (2 * D%(L)) / (TSPAN * Q)
360 Z(L) = R / Q
380  NEXT L
400  REM :DO LEAST SQUARES
401  PRINT "  "
403  PRINT " LINEAR REGRESSION "
405  PRINT "  "
407  PRINT "   "
408 M = LJ
410  FOR L = M TO NW
420 XY = XY + Z(L) * W(L)
430 X = X + W(L)
440 Y = Y + Z(L)
450 XSQ = XSQ + (W(L) ^ 2)
455 YSQ = YSQ + (Z(L) ^ 2)
460  NEXT L
470 G = NNN - M
480 TAU = ((XY * G) - (X * Y)) / ((XSQ * G) - (X * X))
500 AU =  ABS (TAU)
510 HIT = ((XSQ * Y) - (XY * X)) / ((XSQ * G) - (X ^ 2))
511  FOR I = M TO NNN - 1
515 RJ = (Z(I) - (HIT + TAU * W(I))) ^ 2
517 T3 = RJ + T3
518  NEXT I
519  REM  DO RESIDUAL CALCULATIONS
520 U2 = XSQ - (X * X / NNN)
521 V2 = (XSQ - (TAU * TAU * U2)) / (NNN - 2)
522 VSQ = V2 / U2
523 CL = VSQ * T2
525 HTT = HIT * (MD + 0.5) / (AU * ML)
527 CL =  ABS (CL)
530  PRINT "TAU=";AU;" 95%=+/-";CL
535  PRINT "A= ";HIT;"  RESIDUAL= ";T3
590  GOTO 593
591  PRINT "NUMBER OF DATA POINTS EXCEEDED"
592  GOTO 650
593 MY = 0:MX = 0
595  FOR K = M TO NNN
596  IF MY > Z(K) THEN 598
597 MY = Z(K)
598  IF MX > W(K) THEN 600
599 MX = W(K)
600  NEXT K
601  FOR K = M TO NNN
602 Z(K) = Z(K) / MY * 158
603 W(K) = W(K) / MX * 252
604  IF W(K) < 0 THEN W(K) = 0
605 F%(K) =  INT (W(K))
606  IF Z(K) < 0 THEN Z(K) = 0
607 D%(K) =  INT (Z(K))
608  NEXT K
609  REM  PLOT DATA FILES ON SCREEN
610  HGR
612  HPLOT 0,0 TO 0,158 TO 252,158 TO 252,0 TO 0,0
615  FOR K = M TO NNN - 1
616  IF (158 - D%(K)) > 158 THEN 625
617  IF (158 - D%(K)) < 0 THEN 625
```

```
620  HPLOT F%(K),158 - D%(K)
625  NEXT K
650  END

]
```

```
]?""

]LOAD CONVD
]LIST

1   REM
2   REM  THIS PROGRAM CONSTRUCTS A DECAY
3   REM  FILE FROM A KNOWN LAMP FILE.
4   REM  THIS PROGRAM DOES NOT ADD RANDOM
5   REM  TO THE CALCULATED DECAY FILE.
6   REM
10  CALL  - 936
20  DIM G(12),CF(12),L%(1000),DC(1000),S(12)
30 D$ = ""
40  PRINT D$;"NOMON I,O"
50  INPUT "NUMBER OF CONSTANTS...";NK
60  PRINT "THE FITTING CONSTANTS ARE.."
70  FOR I = 1 TO NK
80  PRINT I
90  INPUT G(I)
100  NEXT I
110  PRINT "THE COEFF.S ARE..."
120  FOR I = 1 TO NK
130  PRINT I
140  INPUT CF(I)
150  NEXT I
160  INPUT "LAMP FILE IS ...";A$
161 S1$ = "6,D"
162  INPUT "DRIVE # ";NV$
170  INPUT "CONV'D DATA TO GO IN FILE..";B$
171 S2$ = "6,D"
172  INPUT "DRIVE # ";NY$
200  PRINT D$;"OPEN";A$;",S";S1$;NV$
210  PRINT D$;"READ";A$
220  FOR K = 1 TO 1000
230  INPUT L%(K)
240  NEXT K
250  PRINT D$;"CLOSE"
300  FOR J = 1 TO NK
310 XP(J) =  EXP ( - .097 / G(J))
320  NEXT J
330 DT = .097
340 DC(1) = 0:S(1) = 0
350  FOR I = 2 TO 1000
360 DC(I) = 0
370 I1 = I - 1
380  FOR J = 1 TO NK
390 S(J) = XP(J) * S(J) + DT * (L%(I1) * XP(J) + L%(I)) / 2
400 DC(I) = DC(I) + S(J) * CF(J)
410  NEXT J
420  NEXT I
425  FOR K = 1 TO 1000:DC(K) =  INT (DC(K) + 0.5): NEXT K
430  PRINT D$;"OPEN";B$;",S";S2$;NY$
440  PRINT D$;"WRITE";B$
450  FOR K = 1 TO 1000
460  PRINT DC(K)
470  NEXT K
480  PRINT D$;"CLOSE"
```

```
500 END
]
```

```
]?""

]LOAD CONVRD
]LIST

1  REM
2  REM  THIS PROGRAM CONSTRUCTS A
3  REM  RANDOM DECAY FILE FROM
4  REM  A LAMP FILE. THIS PROGRAM ADDS
5  REM  RANDOM NOISE TO THE CALCULATED
6  REM  DECAY FILE.
7  REM
10  CALL  - 936
20  DIM G( 12 ),CF( 12 ),L%( 1000 ),DC( 1000 ),S( 12 )
30 D$ = ""
40  PRINT D$;"NOMON I,O"
50  INPUT "NUMBER OF CONSTANTS...";NK
60  PRINT "THE FITTING CONSTANTS ARE.."
70  FOR I = 1 TO NK
80  PRINT I
90  INPUT G( I )
100  NEXT I
110  PRINT "THE COEFF.S ARE..."
120  FOR I = 1 TO NK
130  PRINT I
140  INPUT CF( I )
150  NEXT I
160  INPUT "LAMP FILE IS ...";A$
161 S1$ = "6,D"
162  INPUT "DRIVE # ";NV$
170  INPUT "CONV'D DATA TO GO IN FILE..";B$
171 S2$ = "6,D"
172  INPUT "DRIVE # ";NY$
200  PRINT D$;"OPEN";A$;",S";S1$;NV$
210  PRINT D$;"READ";A$
220  FOR K = 1 TO 1000
230  INPUT L%( K )
240  NEXT K
250  PRINT D$;"CLOSE"
300  FOR J = 1 TO NK
310 XP( J ) =  EXP ( - .097 / G( J ))
320  NEXT J
330 DT = .097
340 DC( 1 ) = 0:S( 1 ) = 0
350  FOR I = 2 TO 1000
360 DC( I ) = 0
370 I1 = I - 1
380  FOR J = 1 TO NK
390 S( J ) = XP( J ) * S( J ) + DT * (L%( I1 ) * XP( J ) + L%( I )) / 2
400 DC( I ) = DC( I ) + S( J ) * CF( J )
410  NEXT J
420  NEXT I
422  GOTO 510
430  PRINT D$;"OPEN";B$;",S";S2$;NY$
440  PRINT D$;"WRITE";B$
450  FOR K = 1 TO 1000
455 DC( K ) =  INT (DC( K ) + 0.5)
456  IF DC( K ) < 0 THEN DC( K ) = 0.0
```

```
460  PRINT DC(K)
470  NEXT K
480  PRINT D$;"CLOSE"
500  GOTO 600
510  FOR K = 1 TO 1000
512 SR = 0
515  FOR J = 1 TO 12
520 SR = SR +  RND (5)
525  NEXT
526 SR = SR - 6
530 XR = SR *  SQR (DC(K))
535 XR = XR *  SGN ( COS ( RND (5) * PI))
540 DC(K) = DC(K) + XR
550  NEXT K
590  GOTO 430
600  END

]
```

```
PR#0
]?"

?SYNTAX ERROR
]LOAD WRITEAFILE
]LIST

5  DIM A(1000)
10  CALL  - 936
15 D$ = "": REM : DISK CONTROL
16  PRINT D$;"NOMON I,O"
20  PRINT "**************"
30  PRINT "THIS PROGRAM WRITES FILES ON THE DISK"
40  PRINT "**************"
45  REM
46  REM  INPUT FILE VALUES
47  REM
50  INPUT "FILE NAME YOU WISH TO WRITE? ";A$
55 B$ = A$
60  INPUT "SLOT NUMBER? ";S$
70 A$ = A$ + ",S" + S$ + ",V0"
80  INPUT "HOW MANY POINTS? ";NJ
90  FOR K = 1 TO NJ
100  PRINT K
110  INPUT A(K)
120  NEXT K
121  REM
122  REM  WRITE THE FILE TO DISK
123  REM
130  PRINT D$;"OPEN";A$
140  PRINT D$;"WRITE";B$
150  PRINT NJ
160  FOR K = 1 TO NJ: PRINT A(K): NEXT K
170  PRINT D$;"CLOSE";B$
180  PRINT "THANK YOU!"
190  PRINT D$;"MON I,O"
200  END

]PR#0
```

```
]?"

?SYNTAX ERROR
]LOAD SHIFT
]LIST

5  HOME
6  PRINT "******************"
7  PRINT "PHASE SHIFT PROGRAM"
9  PRINT "******************"
10  DIM Y1(1024),Y2(1024),X(1024)
11 D$ = ""
12  REM
13  REM  INPUT PROGRAM PARAMETERS
14  REM
15  PRINT D$,"NOMON I,O"
30  PRINT
51 DTA = 0.097E - 09
52  INPUT "FREQUENCY USED (MHZ) ? ";MHZ
53 MHZ = 1.0E + 06 * MHZ
54 FR = 1.0 / MHZ
55 SI = 100
56  INPUT "INPUT # OF PTS. ";PT
57  INPUT "ENTER LAMP FILENAME ";A$
58  INPUT "INPUT DRIVE NUMBER ";N$
59 S$ = "S6"
60  INPUT "INPUT DECAY FILENAME ";B$
65  INPUT "DRIVE NUMBER ";E$
70 E$ = ",D" + E$
75 N$ = ",D" + N$
76  REM
77  REM  SET INTEGRATION LIMITS
78  REM
80 ENINE =  INT ((FR / (4.0 * DTA)) + 0.5)
82 LRMX = ( INT (FR / DTA)) + SI
84 SLMX = 999999.
85 PI = 3.14159287
86  REM
87  REM  READ DATA FILES LAMP AND
88 DECAYFILES
89  REM
611  PRINT D$;"OPEN";A$;",V0,";S$;N$
612  PRINT D$;"READ";A$: FOR A = 1 TO PT: INPUT Y1(A): NEXT A: PRINT D$;"C
     LOSE";A$
711  PRINT D$;"OPEN";B$;",V0,";S$;E$
712  PRINT D$;"READ";B$: FOR A = 1 TO PT: INPUT Y2(A): NEXT A: PRINT D$;"C
     LOSE";B$
730  PRINT : PRINT "INTEGRATION CALCULATION": PRINT
750  FOR I = SI TO LRMX
760  IF SLMX > Y1(I) THEN KJ = I
770  IF SLMX > Y1(I) THEN SLMX = Y1(I)
780  NEXT I
785  REM
786  REM  CALCULATE CHANNEL NUMBER TO BEGIN
787  REM  AND END INTEGRATIONS
788  REM
790 KJ = KJ + ENINE
791  PRINT "INTEGRATION BEGINS AT ";KJ
792 A1 = KJ
```

```
]PR#0
]?""


]LOAD LAPLACE4
]LIST

5  HOME
6  PRINT "******************"
7  PRINT "LAPLACE TRANSFORM"
8  PRINT "EXP PROGRAM"
9  PRINT "******************"
10  DIM Y1(1024),Y2(1024),X(1024)
11 D$ = ""
15  PRINT D$,"NOMON I,O"
51 DTA = 0.097
52 BJ = 0.097 / 5.00
55  PRINT : PRINT "PLEASE NOTE THAT THIS PROGRAM IS NOT FINISHED AND PROVI
     DES ONLY APPROX RESULTS": PRINT
56  INPUT "INPUT # OF PTS. ";PT
57  INPUT "ENTER LAMP FILENAME ";A$
58  INPUT "INPUT DRIVE NUMBER ";N$
59 S$ = "S6"
60  INPUT "INPUT DECAY FILENAME ";B$
65  INPUT "DRIVE NUMBER ";E$
70 E$ = ",D" + E$
75 N$ = ",D" + N$
100  GOTO 611
400  FOR K = 1 TO 1000
410  IF ML > Y1(K) THEN 445
420 ML = Y1(K)
445  IF MD > Y2(K) THEN 447
446 MD = Y2(K)
447  NEXT K
450  FOR K = 1 TO 1000
460 KG = Y2(K) * ML / MD + 0.5:Y2(K) =  INT (KG): NEXT K
470  GOTO 800
611  PRINT D$;"OPEN";A$;",V0,";S$;N$
612  PRINT D$;"READ";A$: FOR A = 1 TO PT: INPUT Y1(A): NEXT A: PRINT D$;"C
     LOSE";A$
711  PRINT D$;"OPEN";B$;",V0,";S$;E$
712  PRINT D$;"READ";B$: FOR A = 1 TO PT: INPUT Y2(A): NEXT A: PRINT D$;"C
     LOSE";B$
790  GOTO 400
800  REM :CONTINUE
918 TS = 97.0
920  FOR I = 1 TO 1000
925 X(I) = DTA + X(I - 1)
926 Y1(I) = Y1(I) *  EXP (X(I) / TS):Y2(I) = Y2(I) *  EXP (X(I) / TS)
927  NEXT I
928  FOR I = 990 TO 1000:SX = SX + Y2(I): NEXT I:SX = SX / 10.0
929  PRINT "BEGIN INTEGRATION"
930  FOR I = 1 TO 999
940 REA = 0.5 * (Y1(I + 1) - Y1(I)) * DTA
950 NEA = DTA * Y1(I)
960 ACAL = ACAL + NEA + REA
970  NEXT I
971  PRINT "INTEGRATION 1 COMPLETE"
980  FOR I = 1 TO 999
990 REA = 0.5 * (Y2(I + 1) - Y2(I)) * DTA
```

```
794 B1 = (FR / (4.0 * DTA)) + A1
796 C1 = (FR / (2.0 * DTA)) + A1
798 D1 = (3.0 * FR / (4.0 * DTA)) + A1
800 E1 = (5.0 * FR / (4.0 * DTA)) + A1
802 PF = (FR / DTA) + A1
840 BN = A1:LN = C1
845  GOSUB 930
846  REM  DO LAMP INTEGRATIONS AND RESET PARAMETERS BEFORE INTEGRATION
847 Q1 = ACAL
850 BN = C1:LN = PF
852  GOSUB 930
854 Q2 = ACAL
856 BN = B1:LN = D1
858  GOSUB 930
860 Q3 = ACAL
862 BN = D1:LN = E1
863  GOSUB 930
864 Q4 = ACAL
865  REM  CALCULATE THE PHASE ANGLE FOR LAMP AND DATA FILES
866 T1 =  ATN ((Q3 - Q4) / (Q1 - Q2))
867  FOR I = 1 TO PT:Y1(I) = Y2(I): NEXT I
868 BN = A1:LN = C1
869  REM  DO DECAY INTEGRATIONS AND RESET PARAMETERS BEFORE INTEGRATIONS
870  GOSUB 930
872 Z1 = ACAL
874 BN = C1:LN = PF
876  GOSUB 930
878 Z2 = ACAL
880 BN = B1:LN = D1
882  GOSUB 930
884 Z3 = ACAL
886 BN = D1:LN = E1
888  GOSUB 930
890 Z4 = ACAL
892  GOTO 1200
893  REM
894  REM  CALCULATE THE PHASE ANGLE OF THE DECAY CURVE
895  REM
930 NEA = 0:ACAL = 0:REA = 0
935  FOR I = BN TO LN
940 REA = 0.5 * (Y1(I + 1) - Y1(I)) * DTA
950 NEA = DTA * Y1(I)
960 ACAL = ACAL + NEA + REA
965  NEXT I
980  RETURN
990  REM
991  REM  CALCULATE THE PHASE ANGLE OF THE
992  REM  DECAY CURVE
993  REM
1200 T2 =  ATN ((Z3 - Z4) / (Z1 - Z2))
1202  REM
1203  REM  CALCULATE THE LIFETIME (NSEC)
1204  REM  OF THE DIFFERENCE BETWEEN THE
1205  REM  PHASE ANGLES.
1206  REM
1210 TD =  ABS (T1 - T2)
1220 XL =  TAN (TD) / (2.0 * PI * MHZ)
1225  PRINT
1230  PRINT "LIFETIME IS ";XL;" SEC"
```

```
1000 NEA = DTA * Y2( I )
1010 AEAD = AEAD + NEA + REA
1020  NEXT I
1022  PRINT "INTEGRATION 2 COMPLETE"
1030 IO = AEAD / ACAL
1040  FOR I = 1 TO 1000
1042 SS1 =  EXP ( - BJ * X( I ))
1050 Y1( I ) = Y1( I ) * SS1
1060 Y2( I ) = Y2( I ) * SS1
1070  NEXT I
1071  FOR K = 1 TO 1000
1072  IF MR > Y1( K ) THEN 1074
1073 MR = Y1( K )
1074  IF ME > Y2( K ) THEN 1076
1075 ME = Y2( K )
1076  NEXT K
1077  FOR K = 1 TO 1000
1078 KG = Y2( K ) * ML / ME + 0.5:Y2( K ) =  INT (KG): NEXT K
1079  FOR I = 1 TO 1000:Y1( I ) = Y1( I ) * ML / MR: NEXT I
1080  FOR I = 990 TO 1000
1081 BX = BX + Y2( I )
1082  NEXT I:BX = BX / 10
1085  FOR I = 1 TO 999
1090 REA = 0.5 * (Y1( I + 1) - Y1( I )) * DTA
1100 NEA = DTA * Y1( I )
1110 AFAL = AFAL + NEA + REA
1115  NEXT I: PRINT "INTEGRATION 3 COMPLETE"
1116  FOR I = 1 TO 999
1117 REA = 0.5 * (Y2( I + 1) - Y2( I )) * DTA
1118 NEA = DTA * Y2( I )
1119 ABAD = ABAD + NEA + REA
1120  NEXT I: PRINT "INTEGRATION 4 COMPLETE"
1121 IS = ABAD / AFAL
1122 TI = (IO - IS) / (BJ * IS)
1125 TAU = TI * TS / (TI + TS)
1130 A = IO * (MD + 0.5) / (TI * ML)
1239  PRINT "NOT CORRECTED FOR END POINT"
1240  PRINT "TAU=";TAU,"A=";A
1320 CR = SX * ( EXP ( - (BJ + (1 / TAU)) * TS))
1322 FR = SX * ( EXP ( - (1 / TAU) * TS))
1330 CR = CR / ((BJ + (1 / TAU)) * AFAL)
1332 FR = FR / ((1 / TAU) * ACAL)
1335 IS = IS + CR:IO = IO + FR
1336 TU = TAU
1337 CN = CN + 1
1338 TI = (IO - IS) / (BJ * IS)
1340 TAU = TI * TS / (TI + TS)
1345 A = IO * (MD + 0.5) / (TAU * ML)
1350  PRINT DV
1360 DV =  ABS (TAU - TU)
1370  IF DV < 0.0001 GOTO 1500
1380  IF CN > 40 GOTO 1500
1390  GOTO 1320
1500  PRINT : PRINT "CORRECTION ";CN;" TAU=";TAU;" A=";A

]
```

```
PR#0
]?""

]LOAD TRANS$
]LIST

10 D$ = ""
11  PRINT D$,"NOMON I,O"
20  DIM ZZ$( 2000 )
22  PRINT
25  PRINT "THIS PRGM. TRANSFERS STRING FILES "
26  PRINT "FROM DISK TO DISK": PRINT
29 S$ = "S6"
30  INPUT "FILE NAME ? ";A$
31  INPUT "LENGTH OF FILE? ";L
32  INPUT "DRIVE TRANSFERING FROM ?";E$
33 N$ = "1": IF E$ = "1" THEN N$ = "2"
34 E$ = ",D" + E$
35 N$ = ",D" + N$
36  REM
37  REM  INPUT DATA FROM DISK
38  REM
40  PRINT D$,"OPEN";A$;",V0,";S$;E$
50  PRINT D$,"READ";A$
60  FOR I = 1 TO L: INPUT ZZ$( I ): NEXT I
70  PRINT D$,"CLOSE";A$
71  REM
72  REM  OUTPUT DATA FROM DISK
73  REM
80  PRINT D$,"OPEN";A$;",V0,";S$;N$
90  PRINT D$,"WRITE";A$
100  FOR I = 1 TO L: PRINT ZZ$( I ): NEXT I
110  PRINT D$,"CLOSE";A$
115  PRINT D$,"MON I,O"
120  END

]PR#0
```

```
]?""

]LOAD TLIST$
]LIST

2  HOME
4  PRINT "**************"
6  PRINT "THIS PROGRAM WILL"
8  PRINT "LIST A DATA FILE."
9  PRINT "**************"
10  DIM ZZ$( 2000 )
11 D$ = ""
15  REM  INITIALIZE VARIABLES
20  PRINT D$,"NOMON I,O"
25  PRINT
30  INPUT "ENTER FILE NAME ";A$
35  INPUT "LENGTH OF FILE ? ";X
37  INPUT "WHAT DRIVE NO. ?";N$
38 N$ = ",D" + N$
39 S$ = "S6"
40  REM
41  REM  READ DATA FILE FROM THE DISK
42  REM
45  PRINT D$,"OPEN";A$;",V0,";S$;N$
50  PRINT D$,"READ";A$
60  FOR I = 1 TO X: INPUT ZZ$( I ): NEXT I
70  PRINT D$,"CLOSE",A$
72  REM
73  REM  WRITE DATA FILES OUT IN BLOCKS
74  REM
80  INPUT "DISPLAYED IN BLOCKS ?";A$:A$ =  LEFT$ (A$,1)
100  IF A$ = "Y" THEN 121
110  FOR I = 1 TO X: PRINT I,ZZ$( I ): NEXT I
120  GOTO 230
121  PRINT " # OF PTS. PER BLOCK ?"
122  INPUT "( 20 FIT ON SCREEN )";N
123  FLASH : PRINT "TO BREAK TYPE B": NORMAL
125 R =  INT (X / N) + 1
130  FOR I = 1 TO R
150  FOR J = (I - 1) * N + 1 TO I * N
160  PRINT J,ZZ$( J )
170  NEXT J
180  PRINT "RETURN FOR NEXT BLOCK       ";"PAGE ";I;" OF ";R
190  GET A$
191  HOME
195  IF A$ = "B" THEN 220
200  NEXT I
205  GOTO 230
206  REM
207  REM  EXIT SUBROUTINE
208  REM
220  PRINT "BREAK TO END"
230  INPUT "DO YOU WANT TO LIST AGAIN ? ";LS$:LS$ =  LEFT$ (LS$,1)
240  IF LS$ = "Y" THEN 80
250  PRINT D$,"MON I,O"

]PR#0
```

```
222  PRINT D$,"OPEN";A$;",V0,";S$;N$
223  PRINT D$,"WRITE";A$
224  FOR I = 1 TO X: PRINT ZZ( I ): NEXT I
226  PRINT D$,"CLOSE";A$
227  FOR I = 1 TO X:R( I ) = ZZ( I ): NEXT I
230  INPUT "DO YOU WANT TO LIST AGAIN ? ";LS$:LS$ =  LEFT$ (LS$,1)
240  IF LS$ = "Y" THEN 145
245  IF LZ$ = "Y" THEN 300
250  INPUT "ARE THESE VALUES CORRECT ? ";LZ$:LZ$ =  LEFT$ (LZ$,1)
260  IF LZ$ = "Y" THEN 222
300  PRINT D$,"MON I,O"

]PR#0
```

```
]?""

]LOAD SEND
]LIST

4  HOME
5  PRINT "****************"
6  PRINT "APPLE/IBM 370"
7  PRINT "DATA TRANSFER"
8  PRINT "PROGRAM"
9  PRINT "****************"
10  DIM ITEM( 1024 )
15  REM
16  REM  INPUT PROGRAM PARAMETER
17  REM
20 D$ = ""
21  REM  "CONTROL-D"
25  PRINT D$;"NOMON C,I,O"
26  PRINT
30  INPUT "WHAT FILE NAME ON APPLE ?";A$
40  INPUT "NUMBER OF DATA POINTS ?";NK
60  INPUT "DRIVE NUMBER ?";D1$
65 S1$ = "S6,D" + D1$
66  REM
67  REM  READ DATA FILES TO TRANSFER TO THE IBM 370
68  REM
70  PRINT D$;"OPEN";A$;",V0,";S1$
80  PRINT D$;"READ";A$
90  FOR I = 1 TO NK: INPUT ITEM( I ): NEXT I
91  PRINT D$;"CLOSE"
92 AJ = NK / 20: HOME
93  FOR I = 1 TO AJ: PRINT "TIME( NSEC )      INTENSITY    PAGE ";I;" OF ";AJ:
     M = I * 20:J = M + 20: FOR N = M TO J
94 DE = DE + 0.097:DE$ =  LEFT$ ( STR$ (DE ),4): PRINT DE$,ITEM( N ): NEXT N
95  FOR L = 1 TO 300: NEXT L: HOME : NEXT I
96  REM  DATA TRANSFER ROUTINE
97  FLASH : PRINT "DATA TRANSFER": NORMAL
98 BIG = 999999
99 C$ = "":G$ = ""
100  REM  STRING DATA VALUES TOGETHER
101  FOR I = 1 TO NK
103  IF ITEM( I ) > BIG THEN 180
105 ITEM( I ) = ITEM( I ) + 1000000
107  NEXT I
109  PRINT D$;"PR#1"
110  FOR I = 1 TO 10: IF I > NK THEN 185
111 C$ =  STR$ ( ITEM( I )):G$ = G$ + C$: NEXT I
112  PRINT G$
114  PRINT D$;"IN#1"
115  FOR I = 1 TO 5000: NEXT I
116 C$ = "":G$ = ""
117 I = 1
119 I = I + 10
120 M = I + 9
123  FOR N = I TO M
124  IF N > NK THEN 185
125 C$ =  STR$ ( ITEM( N ))
126 G$ = G$ + C$
```

```
127  NEXT N
130  PRINT G$
133  PRINT D$;"PR#0"
152  GET B$
155  PRINT B$
158  REM :CONTROL-S
159  REM  INPUT ROUTINE TO RECEIVE LINE FEED
160  IF B$ >  < "" THEN 152
161  REM :CONTROL-S
163  PRINT D$;"PR#1"
165 C$ = "":G$ = ""
170  GOTO 119
180  PRINT D$;"PR#0"
181  FLASH
182  PRINT "***ERROR-INPUT GREATER THEN 999,999 ***"
184  NORMAL : GOTO 225
185  PRINT G$
186  PRINT D$;"PR#0"
187  GET B$
189  PRINT B$
191  IF B$ >  < "" THEN 187
192  REM :CONTRL-S
193 S =  - 16336
194  PRINT D$;"PR#0"
195 V =  PEEK (S) -  PEEK (S) +  PEEK (S) -  PEEK (S) +  PEEK (S) -  PEEK
     (S) +  PEEK (S) -  PEEK (S):I = I + 1
200  FOR L = 1 TO 20: NEXT L
202  IF I < 20 GOTO 195
205  FOR I = 1 TO 20
207 V =  PEEK (S) -  PEEK (S) +  PEEK (S)
208  NEXT I
209 JJ = JJ + 1
210  IF JJ < 10 GOTO 205
211  REM  END OF DATA TRANSFER ROUTINE
230  PRINT "TO BREAK TYPE :"
231  FLASH
232  PRINT "CNTRL-A  CNTRL-S  RETURN"
234  NORMAL
250  END

]PR#0
```

```
]?""

]LOAD FIX
]LIST

2  HOME
4  PRINT "**************"
6  PRINT "THIS PROGRAM WILL FIX"
8  PRINT "DAMAGED DATA FILES."
9  PRINT "**************"
10  DIM ZZ( 2000 ),R( 2000 )
11 D$ = ""
20  PRINT D$,"NOMON I,O"
25  PRINT
30  INPUT "ENTER FILE NAME ";A$
35  INPUT "LENGTH OF FILE ? ";X
37  INPUT "WHAT DRIVE NO. ?";N$
38 N$ = ",D" + N$
39 S$ = "S6"
40  PRINT D$,"OPEN";A$;",V0,";S$;N$
50  PRINT D$,"READ";A$
60  FOR I = 1 TO X: INPUT ZZ( I ): NEXT I
70  PRINT D$,"CLOSE";A$
75 ZZ( 1 ) = 0
80  FOR I = 1 TO X
85 R( I ) = ZZ( I )
90 AL = ZZ( I )
100 C = ZZ( I + 1 ):D = ZZ( I + 2 )
110  IF D > C THEN 120
115  GOTO 140
120  IF C > AL THEN 130
125  GOTO 140
130  IF ZZ( I ) > M THEN M = ZZ( I )
140  NEXT I
141  FOR I = 1 TO X
142 LL =  ABS ( ZZ( I ) - ZZ( I - 1 ))
143  IF LL > M THEN ZZ( I ) = ZZ( I + 1 )
144  NEXT I
145  INPUT "DATA DISPLAYED IN BLOCKS ? ";E$:E$ =  LEFT$ (E$,1)
146  IF E$ = "Y" THEN 149
147  FOR I = 1 TO X: PRINT I,R( I ),ZZ( I ): NEXT I
148  GOTO 230
149  PRINT "NUMBER OF PTS. PER BLOCK ?"
150  INPUT "( 20 FIT ON SCREEN )";N
151  FLASH : PRINT "TO BREAK TYPE B": NORMAL
153 R =  INT (X / N) + 1
155  FOR I = 1 TO R
156  FOR J = (I - 1) * N + 1 TO I * N
160  PRINT J,R( J ),ZZ( J )
170  NEXT J
180  PRINT "RETURN FOR NEXT BLOCK      ";"PAGE ";I;" OF ";R
190  GET F$
191  HOME
195  IF F$ = "B" THEN 220
200  NEXT I
205  GOTO 230
220  PRINT "BREAK TO END"
221  GOTO 230
```

```
]LIST LOGP

5  HOME
6  PRINT "******************"
7  PRINT "PLOT LOG PROGRAM"
8  PRINT "******************"
10  DIM Y2(1024)
11 D$ = ""
12  REM
13  REM  INPUT PARAMETERS
14  REM
15  PRINT D$,"NOMON I,O"
55  PRINT
56  INPUT "INPUT # OF PTS. ";PT
59 S$ = "S6"
60  INPUT "INPUT DECAY FILENAME ";B$
65  INPUT "DRIVE NUMBER ";E$
70 E$ = ",D" + E$
71  REM
72  REM  READ DATA FILE TO PLOT
73  REM
711  PRINT D$;"OPEN";B$;",VO,";S$;E$
712  PRINT D$;"READ";B$: FOR A = 1 TO PT: INPUT Y2(A): NEXT A: PRINT D$;"C
     LOSE";B$
?00  REM
?01  REM  CALCULATE THE LOG AND SCALE
?02  REM  POINTS
?03  REM
920  FOR J = 1 TO PT
 ?2  IF Y2(J) < 1 THEN Y2(J) = 1
924 Y2(J) = 2.303 *  LOG (Y2(J))
?28  IF MY > Y2(J) THEN 930
 ?9 MY = Y2(J)
     NEXT J
    MX = PT * 0.097
9?   FOR K = 1 TO PT
    ?2(K) = Y2(K) / MY * 157
9.   ?2(K) =  INT (Y2(K))
?5?  ?EXT K
960 S = 0.1
?41  ?EM
9?2  REM  PLOT DATA POINTS ROUTINE
963  REM
965  HGR2
966  HPLOT 0,0 TO 0,158 TO 256,158 TO 256,0 TO 0,0
970  FOR I = 1 TO PT
972 AA = AA + 0.097
973 X =  INT (AA + 0.097) / MX * 256
975  HPLOT X,158 - Y2(I)
979  IF X < S * 256 THEN 985
980  HPLOT X,0.0 TO X,10
983 S = S + 0.1
985  NEXT I
990  PRINT D$,"MON I,O"
995  PR# 0
```

```
996  PRINT ""
1000  END

]PR#0
```

```
]LOAD STD
]LIST

5  HOME
10  DIM X(100)
15  PRINT "******************"
20  PRINT "THIS PROGRAM CALCULATES"
30  PRINT "THE STANDARD DEVIATION"
40  PRINT "OF SEVERAL LIFETIME RUNS"
45  PRINT "******************"
46  REM
47  REM  INPUT DATA VALUES
48  REM
50  PRINT
60  INPUT "NUMBER OF RUNS ? ";NN
70  FOR N = 1 TO NN
80  PRINT "INPUT DATAPOINT ";N
90  INPUT X(N)
100  NEXT N
101  HOME
102  PRINT "RUN NO.      LIFETIME     X**2"
103  PRINT
104  REM
105  REM  CALCULATE THE RESIDUALS AND
106  REM  CHI SQUARED
107  REM
110  FOR I = 1 TO NN
120 RS = X(I) ^ 2
130  PRINT I;"    ";X(I);"     ";RS
140 TR = TR + RS
150 NR = NR + X(I)
160  NEXT I
170 STD = (TR - (NR * NR / NN)) / (NN - 1)
180 STD =  SQR (STD)
185  PRINT "S OF X=";NR;"  S OF X**2=";TR
186 STD =  INT (STD * 1000 + 0.5)
187  REM
188  REM  PRINT VALUES
189  REM
190  PRINT "THE STANDARD DEVIATION=";STD;" PSEC"
200 NR = NR / NN
205 STD = STD / 1000
210 TP = 100 * STD / NR
220  PRINT "THE AVERAGE LIFETIME IS= ";NR
230  PRINT "THE PERCENT RSD IS = ";TP

]PR#0
```

```
]?""

]LOAD LAPLACE
]LIST

5  HOME
6  PRINT "******************"
7  PRINT "LAPLACE TRANSFORM"
8  PRINT "EXP PROGRAM"
9  PRINT "******************"
10  DIM Y1( 1024 ),Y2( 1024 ),X( 1024 )
11 D$ = ""
15  PRINT D$,"NOMON I,O"
51 DTA = 0.097
55  PRINT : PRINT "PLEASE NOTE THAT THIS PROGRAM IS NOT FINISHED AND PROVI
    DES ONLY APPROX RESULTS": PRINT
56  INPUT "INPUT # OF PTS. ";PT
57  INPUT "ENTER LAMP FILENAME ";A$
58  INPUT "INPUT DRIVE NUMBER ";N$
59 S$ = "S6"
60  INPUT "INPUT DECAY FILENAME ";B$
65  INPUT "DRIVE NUMBER ";E$
70 E$ = ",D" + E$
75 N$ = ",D" + N$
100  GOTO 611
400  FOR K = 1 TO 1000
410  IF ML > Y1(K) THEN 445
420 ML = Y1(K)
445  IF MD > Y2(K) THEN 447
446 MD = Y2(K):LJ = K
447  NEXT K
450  FOR K = 1 TO 1000
460 KG = Y2(K) * ML / MD + 0.5:Y2(K) =  INT (KGG): NEXT K
465  FOR I = 990 TO 1000:SX = SX + Y2(I): NEXT I:SX = SX / 10.0
470  GOTO 800
611  PRINT D$;"OPEN";A$;",V0,";S$;N$
612  PRINT D$;"READ";A$: FOR A = 1 TO PT: INPUT Y1(A): NEXT A: PRINT D$;"C
     LOSE";A$
711  PRINT D$;"OPEN";B$;",V0,";S$;E$
712  PRINT D$;"READ";B$: FOR A = 1 TO PT: INPUT Y2(A): NEXT A: PRINT D$;"C
     LOSE";B$
790  GOTO 400
800  REM :CONTINUE
918 TS = 97.0
920  FOR I = 1 TO 1000
925 X(I) = DTA + X(I - 1)
926 Y1(I) = Y1(I) *  EXP (X(I) / TS):Y2(I) = Y2(I) *  EXP (X(I) / TS)
927  NEXT I
928  PRINT "BEGIN INTEGRATION"
930  FOR I = 1 TO 999
940 REA = 0.5 * (Y1(I + 1) - Y1(I)) * DTA
950 NEA = DTA * Y1(I)
960 ACAL = ACAL + NEA + REA
970  NEXT I
971  PRINT "INTEGRATION 1 COMPLETE"
980  FOR I = 1 TO 999
990 REA = 0.5 * (Y2(I + 1) - Y2(I)) * DTA
1000 NEA = DTA * Y2(I)
```

```
1010 AEAD = AEAD + NEA + REA
1020  NEXT I
1022  PRINT "INTEGRATION 2 COMPLETE"
1030 IO = AEAD / ACAL
1040  FOR I = 1 TO 1000
1042 SS1 =  EXP ( - 0.02 * X(I))
1050 Y1(I) = Y1(I) * SS1
1060 Y2(I) = Y2(I) * SS1
1070  NEXT I
1080  FOR I = 1 TO 999
1090 REA = 0.5 * (Y1(I + 1) - Y1(I)) * DTA
1100 NEA = DTA * Y1(I)
1110 AFAL = AFAL + NEA + REA
1115  NEXT I: PRINT "INTEGRATION 3 COMPLETE"
1116  FOR I = 1 TO 999
1117 REA = 0.5 * (Y2(I + 1) - Y2(I)) * DTA
1118 NEA = DTA * Y2(I)
1119 ABAD = ABAD + NEA + REA
1120  NEXT I: PRINT "INTEGRATION 4 COMPLETE"
1121 IS = ABAD / AFAL
1122 TI = (IO - IS) / (0.02 * IS)
1125 TAU = TI * TS / (TS + TI)
1130 A = IO * (MD + 0.5) / (TI * ML)
1239  PRINT "NOT CORRECTED FOR END POINT"
1240  PRINT "TAU=";TAU,"A=";A
1320 CR = SX * ( EXP ( - (0.02 + (1 / TAU)) * TS))
1330 CR = CR / ((0.02 + (1 / TAU)) * AFAL)
1335 IS = IS + CR
1336 TU = TAU
1337 CN = CN + 1
1338 TI = (IO - IS) / (0.02 * IS)
1340 TAU = TI * TS / (TS + TI)
1345 A = IO * (MD + 0.5) / (TI * ML)
1350  PRINT : PRINT "CORRECTION ";CN;" TAU=";TAU;" A=";A
1360 DV =  ABS (TAU - TU)
1370  IF DV = 0.0 GOTO 1500
1380  IF CN > 40 GOTO 1500
1390  GOTO 1320
1500  END

]
```

```
]?""

]LOAD SCATTERED
]LIST

2  HOME
4  PRINT "***************"
6  PRINT "THIS PROGRAM WILL ELIMINATE"
8  PRINT "SCATTERED LIGHT."
9  PRINT "***************"
10  DIM ZZ( 2000 ),R( 2000 )
11 D$ = ""
20  PRINT D$,"NOMON I,O"
25  PRINT
30  INPUT "ENTER FILE NAME ";A$
35  INPUT "LENGTH OF FILE ? ";X
37  INPUT "WHAT DRIVE NO. ?";N$
38 N$ = ",D" + N$
39 S$ = "S6"
40  PRINT D$,"OPEN";A$;",VO,";S$;N$
50  PRINT D$,"READ";A$
60  FOR I = 1 TO X: INPUT ZZ( I ): NEXT I
70  PRINT D$,"CLOSE";A$
80  FOR I = 30 TO 90
85 Q = Q + ZZ( I ): NEXT I
90 Q = Q / 60
100  FOR I = 1 TO X
101 R( I ) = ZZ( I ):ZZ( I ) = ZZ( I ) - Q
102  IF ZZ( I ) < 0 THEN ZZ( I ) = 0
103 ZZ( I ) =  INT (ZZ( I ))
104  NEXT I
145  INPUT "DATA DISPLAYED IN BLOCKS ? ";E$:E$ =  LEFT$ (E$,1)
146  IF E$ = "Y" THEN 149
147  FOR I = 1 TO X: PRINT I,R( I ),ZZ( I ): NEXT I
148  GOTO 230
149  PRINT "NUMBER OF PTS. PER BLOCK ?"
150  INPUT "( 20 FIT ON SCREEN )";N
151  FLASH : PRINT "TO BREAK TYPE B": NORMAL
153 R =  INT (X / N) + 1
155  FOR I = 1 TO R
156  FOR J = (I - 1) * N + 1 TO I * N
160  PRINT J,R( J ),ZZ( J )
170  NEXT J
180  PRINT "RETURN FOR NEXT BLOCK      ";"PAGE ";I;" OF ";R
190  GET F$
191  HOME
195  IF F$ = "B" THEN 220
200  NEXT I
205  GOTO 230
220  PRINT "BREAK TO END"
221  GOTO 230
222  PRINT D$,"OPEN";A$;",VO,";S$;N$
223  PRINT D$,"WRITE";A$
224  FOR I = 1 TO X: PRINT ZZ( I ): NEXT I
226  PRINT D$,"CLOSE";A$
227  FOR I = 1 TO X:R( I ) = ZZ( I ): NEXT I
230  INPUT "DO YOU WANT TO LIST AGAIN ? ";LS$:LS$ =  LEFT$ (LS$,1)
240  IF LS$ = "Y" THEN 145
```

```
245  IF LZ$ = "Y" THEN 300
250  INPUT "ARE THESE VALUES CORRECT ? ";LZ$:LZ$ =  LEFT$ (LZ$,1)
255  IF LZ$ = "Y" THEN  PRINT "INPUT FILENAME "
256  IF LZ$ = "Y" THEN  INPUT A$
260  IF LZ$ = "Y" THEN 222
300  PRINT D$,"MON I,O"

]PR#0
```

```
]LOAD PLOT1
]LIST

5  HOME
6  PRINT "******************"
7  PRINT "PLOT DATA PROGRAM"
8  PRINT "******************"
9  REM  THIS PROGRAM WILL PLOT 1 DATA FILE ON THE SCREEN
10  DIM Y1(1024),Y2(1024)
11 D$ = ""
12  REM
13  REM  INPUT PARAMETERS
14  REM
15  PRINT D$,"NOMON I,O"
50 T = .01
53  PRINT
55  INPUT "INPUT BEGINNING POINT PLOTTED ? ";RN
56  INPUT "INPUT ENDING POINT PLOTTED ? ";PT
57  INPUT "ENTER DATA FILENAME ";A$
58  INPUT "INPUT DRIVE NUMBER ";N$
59 S$ = "S6"
75 N$ = ",D" + N$
500  REM
501  REM  READ DATA VALUES
502  REM
611  PRINT D$;"OPEN";A$;",V0,";S$;N$
612  PRINT D$;"READ";A$: FOR A = 1 TO PT: INPUT Y1(A): NEXT A: PRINT D$;"C
     LOSE";A$
900  REM
901  REM  SCALE DATA VALUES
902  REM
920  FOR J = RN TO PT
923  IF MZ > Y1(J) THEN 930
924 MZ = Y1(J)
930  NEXT J
931 MX = (PT - RN) * 0.097
935  FOR K = RN TO PT
946 Y1(K) = Y1(K) / MZ * 157
950  IF Y1(K) < 0 THEN Y1(K) = 0
951 Y1(K) =  INT (Y1(K))
959  NEXT K
960 S = 0.1
961 LN = RN * 0.097
962 AA = (RN - 1) * 0.097
963  REM   PLOT DATA VALUES ON THE SCREEN
964  REM
965  HGR2
966  HPLOT 0,0 TO 0,158 TO 256,158 TO 256,0 TO 0,0
970  FOR I = RN TO PT
972 AA = AA + 0.097
973 X =  INT ((AA + 0.097 - LN) / MX * 256)
974  IF X > 256 THEN X = 256
975  HPLOT X,158 - Y1(I)
979  IF X < S * 256 THEN 985
980  HPLOT X,0.0 TO X,10
983 S = S + 0.1
```

```
985  NEXT I
990  PRINT D$;"MON I,O"
995  PR# 0
996  PRINT ""
1000  GET Z$: TEXT : HOME : END

]PR#0
```

```
]?""

]LOAD CURVESM
5  HOME
6  PRINT "******************"
7  PRINT "CURVE SMOOTHING PROGRAM"
8  PRINT "******************"
10  DIM Y1(1024)
11 D$ = ""
12  REM
13  REM  THIS PROGRAM WILL MANUALLY
14  REM  SMOOTH A DATA CURVE.
15  PRINT D$,"NOMON I,O"
16  PRINT
17  GOTO 1300
18  PRINT
50 T = .01
51  INPUT "TOTAL FILE LENGTH ? ";RT
52  INPUT "INPUT FIRST POINT PLOTTTED ? ";RN
55  PRINT
56  INPUT "LAST POINT TO BE PLOTTED ? ";PT
57  INPUT "ENTER DATA FILENAME ";A$
58  INPUT "INPUT DRIVE NUMBER ";N$
59 S$ = "S6"
75 N$ = ",D" + N$
611  PRINT D$;"OPEN";A$;",V0,";S$;N$
612  PRINT D$;"READ";A$: FOR A = 1 TO RT: INPUT Y1(A): NEXT A: PRINT D$;"C
     LOSE";A$
920  FOR J = RN TO PT
923  IF MZ > Y1(J) THEN 930
924 MZ = Y1(J)
930  NEXT J
931 MX = (PT - RN) * 0.097
935  FOR K = RN TO PT
946 Y1(K) = Y1(K) / MZ * 157
950  IF Y1(K) < 0 THEN Y1(K) = 0
951 Y1(K) =  INT (Y1(K))
959  NEXT K
960 S = 0.1
961 AA = (RN - 1) * 0.097
962 LN = RN * 0.097
965  HGR2
966  HPLOT 0,0 TO 0,158: HPLOT 256,158 TO 256,0
967  HCOLOR= 7
970  FOR I = RN TO PT
972 AA = AA + 0.097
973 X =  INT ((AA + 0.097 - LN) / MX * 256)
974  IF X > 256 THEN X = 256
975  HPLOT X,158 - Y1(I)
979  IF X < S * 256 THEN 985
983 S = S + 0.1
985  NEXT I
986 X = 0
987 BN = RN
990  PRINT "": INPUT EA$
991  IF EA$ = "E" GOTO 1550
```

```
992  IF EA$ = "C" GOTO 1099
993  IF EA$ = "C*" GOTO 1150
994  IF EA$ = "S" THEN 1500
995  PRINT "": INPUT AL
996 NN = 256 * AL / (PT - RN)
997  HCOLOR= 0: HPLOT X,0.0 TO X,158: HCOLOR= 7
1007  HPLOT X,158 - Y1(BN)
1010  IF EA$ = "A" THEN X = X + NN
1015  IF EA$ = "A" THEN BN = AL + BN
1020  IF EA$ = "B" THEN X = X - NN
1025  IF EA$ = "B" THEN BN = BN - AL
1030  HPLOT X,0.0 TO X,158
1040  GOTO 990
1099 BN = BN - 1
1100 Y1(BN) = (Y1(BN - 1) + Y1(BN + 1)) / 2
1105  PRINT ""
1110  GOTO 961
1150 BN = BN - 1
1160 Y1(BN) = (Y1(BN - 3) + Y1(BN + 3)) / 2
1161  PRINT ""
1170  GOTO 961
1300  PRINT "AFTER THE BELL TYPE:"
1310  PRINT "    C*-CHANGE ALL"
1320  PRINT "    C -CHANGE"
1330  PRINT "    E -END EDIT"
1340  PRINT "    A -ADVANCE POINTER"
1350  PRINT "    B -RETRIEVE POINTER"
1355  PRINT "    S -SAVE DATA"
1360  PRINT
1370  PRINT "AFTER TWO BELLS TYPE:"
1380  PRINT " A NUMBER VALUE TO MOVE"
1390  PRINT " THE POINTER"
1400  PRINT
1410  GOTO 50
1500  PRINT D$;"OPEN";A$;",V0,";S$;N$
1515  PRINT D$;"WRITE";A$: FOR A = 1 TO RT: PRINT Y1(A): NEXT A: PRINT D$;
     "CLOSE";A$
1550  TEXT
1560  END

]

PR#0
```

```
]LIST

5  HOME
6  PRINT "******************"
7  PRINT "PHASE SHIFT PROGRAM"
9  PRINT "******************"
10  DIM Y1( 1024 ),Y2( 1024 ),X( 1024 )
11 D$ = ""
12  REM
13  REM  INPUT PROGRAM PARAMETERS
14  REM
15  PRINT D$,"NOMON I,O"
30  PRINT
51 DTA = 0.09999876E - 09
52  INPUT "FREQUENCY USED (MHZ) ? ";MHZ
53 MHZ = 1.0E + 06 * MHZ
54 FR = 1.0 / MHZ
55 SI = 100 - ( INT (FR / DTA))
56  INPUT "INPUT # OF PTS. ";PT
57  INPUT "ENTER LAMP FILENAME ";A$
58  INPUT "INPUT DRIVE NUMBER ";N$
59 S$ = "S6"
60  INPUT "INPUT DECAY FILENAME ";B$
65  INPUT "DRIVE NUMBER ";E$
70 E$ = ",D" + E$
75 N$ = ",D" + N$
76  REM
77  REM  SET INTEGRATION LIMITS
78  REM
80 ENINE =  INT ((FR / (4.0 * DTA)) + 0.5)
82 LRMX = ( INT (FR / DTA)) + SI
85 PI = 3.14159287
90 NPER = 4
95  IF MHZ > 50.0E + 06 THEN 611
97 NPER = 2
98  REM
99  REM  READ DATA FILES (LAMP AND DECAY)
100  REM
611  PRINT D$;"OPEN";A$;",VO,";S$;N$
612  PRINT D$;"READ";A$: FOR A = 1 TO PT: INPUT Y1(A): NEXT A: PRINT D$;"C
     LOSE";A$
650  HOME
660  PRINT "LAMP=";A$;" DECAY=";B$;" MHZ=";MHZ / 1.0E + 06
670  PRINT
711  PRINT D$;"OPEN";B$;",VO,";S$;E$
712  PRINT D$;"READ";B$: FOR A = 1 TO PT: INPUT Y2(A): NEXT A: PRINT D$;"C
     LOSE";B$
713  REM
714  REM  CALCULATE CHANNEL NUMBERS TO BEGIN AND
715  REM  END INTEGRATIONS.
716  REM
721  FOR DD = 1 TO NPER
722 SI = SI + ( INT (FR / DTA))
723 LRMX = LRMX + ( INT (FR / DTA))
724 SLMX = 999999.9
```

```
750  FOR I = ST TO LPMX
760  IF SLMX > Y1(I) THEN KJ = I
770  IF SLMX > Y1(I) THEN SLMX = Y1(I)
780  NEXT I
790 KJ = KJ + ENINE
792 A1 = KJ
794 B1 = (FR / (4.0 * DTA)) + A1
796 C1 = (FR / (2.0 * DTA)) + A1
798 D1 = (3.0 * FR / (4.0 * DTA)) + A1
800 E1 = (5.0 * FR / (4.0 * DTA)) + A1
802 PF = (FR / DTA) + A1
803  REM
804  REM  DO LAMP INTEGRATIONS AND RESET
805  REM  INTEGRAL PARAMETERS BEFORE INTEGRATION
806  REM
840 BN = A1:LN = C1
845  GOSUB 930
847 Q1 = ACAL
850 BN = C1:LN = PF
852  GOSUB 930
854 Q2 = ACAL
856 BN = B1:LN = D1
858  GOSUB 930
860 Q3 = ACAL
862 BN = D1:LN = E1
863  GOSUB 930
864 Q4 = ACAL
865 A5 = A5 + Q1:A2 = A2 + Q2:A3 = A3 + Q3:A4 = A4 + Q4
866  REM  CALCULATE THE PHASE ANGLE FOR A LAMP DATA FILE
867 T1 =  ATN ((Q3 - Q4) / (Q1 - Q2))
868 BN = A1:LN = C1
870  GOSUB 981
872 Z1 = ACAL
874 BN = C1:LN = PF
876  GOSUB 981
878 Z2 = ACAL
880 BN = B1:LN = D1
882  GOSUB 981
884 Z3 = ACAL
886 BN = D1:LN = E1
888  GOSUB 981
890 Z4 = ACAL
891 B5 = B5 + Z1:B2 = B2 + Z2:B3 = B3 + Z3:B4 = B4 + Z4
892  GOTO 1200
893  REM
894  REM  INTEGRATION ROUTINE
895  REM
930 NEA = 0:ACAL = 0:REA = 0
935  FOR I = BN TO LN
940 REA = 0.5 * (Y1(I + 1) - Y1(I)) * DTA
950 NEA = DTA * Y1(I)
960 ACAL = ACAL + NEA + REA
965  NEXT I
980  RETURN
981 NEA = 0:ACAL = 0:REA = 0: FOR I = BN TO LN
982 REA = 0.5 * (Y2(I + 1) - Y2(I)) * DTA
983 NEA = DTA * Y2(I)
984 ACAL = ACAL + NEA + REA
985  NEXT I
986  RETURN
```

```
1000  REM
1001  REM  CALCULATE THE PHASE ANGLE OF THE DECAY CURVE
1200 T2 =  ATN ((Z3 - Z4) / (Z1 - Z2))
1210 TD =  ABS (T1 - T2)
1215  REM  CALCULATE THE LIFETIME (NSEC)
1216  REM  OF THE DIFFERENCE BETWEEN THE PHASE ANGLES
1220 XL =  TAN (TD) / (2.0 * PI * MHZ)
1225  PRINT
1230  PRINT "TAU";DD;" = ";XL;" SEC"
1232  REM
1234  REM  CALCULATE THE LIFETIME (NSEC)
1236  REM  OVER THE SEVERAL PERIODS OF
1238  REM  OSCILLATIONS
1240  REM
1250  NEXT DD
1260 C9 =  ATN ((B3 - B4) / (B5 - B2))
1262 C8 =  ATN ((A3 - A4) / (A5 - A2))
1264 C7 =  ABS (C8 - C9)
1265  PRINT : PRINT
1266 SH =  TAN (C7) / (2.0 * PI * MHZ)
1270  PRINT "THE LIFETIME IS ... ";SH

]PR#0
```

```
]?""

]LOAD PLOT2
]LIST

5  HOME
6  PRINT "******************"
7  PRINT "PLOT DATA PROGRAM"
8  PRINT "******************"
9  REM  THIS PROGRAM WILL PLOT 2 DATA FILES ON THE SCREEN
10  DIM Y1( 1024 ),Y2( 1024 )
11 D$ = ""
12  REM
13  REM  INPUT PARAMETERS
14  REM
15  PRINT D$,"NOMON I,O"
50 T = .01
53  PRINT
55  INPUT "INPUT BEGINNING POINT PLOTTED ? ";RN
56  INPUT "INPUT ENDING POINT PLOTTED ? ";PT
57  INPUT "ENTER LAMP FILENAME ";A$
58  INPUT "INPUT DRIVE NUMBER ";N$
59 S$ = "S6"
60  INPUT "INPUT DECAY FILENAME ";B$
65  INPUT "DRIVE NUMBER ";E$
70 E$ = ",D" + E$
75 N$ = ",D" + N$
76  REM
77  REM  READ DATA VALUES FOR 2 FILES
78  REM
611  PRINT D$;"OPEN";A$;",V0,";S$;N$
612  PRINT D$;"READ";A$: FOR A = 1 TO PT: INPUT Y1(A): NEXT A: PRINT D$;"C
     LOSE";A$
711  PRINT D$;"OPEN";B$;",V0,";S$;E$
712  PRINT D$;"READ";B$: FOR A = 1 TO PT: INPUT Y2(A): NEXT A: PRINT D$;"C
     LOSE";B$
900  REM
901  REM  SCALE DATA VALUES FOR 2 PLOTS
902  REM
920  FOR J = RN TO PT
923  IF MZ > Y1(J) THEN 928
924 MZ = Y1(J)
928  IF MY > Y2(J) THEN 930
929 MY = Y2(J)
930  NEXT J
931 MX = (PT - RN) * 0.097
935  FOR K = RN TO PT
945 Y2(K) = Y2(K) / MY * 157
946 Y1(K) = Y1(K) / MZ * 157
950  IF Y1(K) < 0 THEN Y1(K) = 0
951 Y1(K) =  INT (Y1(K))
955  IF Y2(K) < 0 THEN Y2(K) = 0
956 Y2(K) =  INT (Y2(K))
959  NEXT K
960 S = 0.1
961 LN = RN * 0.097
962 AA = (RN - 1) * 0.097
963  REM  PLOT OF DATA VALUES ON THE SCREEN
```

```
[illegible]  REM  IN HIGH RESOLUTION GRAPHICS.
965  HGR2
966  HPLOT 0,0 TO 0,158 TO 256,158 TO 256,0 TO 0,0
970  FOR I = RN TO PT
972 AA = AA + 0.097
973 X =  INT ((AA + 0.097 - LN) / MX * 256)
974  IF X > 256 THEN X = 256
975  HPLOT X,158 - Y1(I)
976  HPLOT X,158 - Y2(I)
979  IF X < S * 256 THEN 985
980  HPLOT X,0.0 TO X,10
983 S = S + 0.1
985  NEXT I
990  PRINT D$,"MON I,O"
995  PR# 0
996  PRINT ""
1000  GET Z$: TEXT : HOME : END

]PR#0
```

Master

The program master sends a lamp and decay file, and receives a residual and autocorrelation file. Accessing OKLA3 is done through the command list program COR.

Table XIX gives a description and explanation of the steps involved.

TABLE XIX

DESCRIPTION AND EXPLANATION STEPS

Line Numbers	Comments
1 - 200	Input data to "basic" program
200 - 286	LOGON procedure
286 - 400	Data transfer control variables
401 - 615	Data sending subroutine
700 - 830	Data input subroutine
834 - 868	Input variables for program execution (5 variables)
900 - 925	Input variable for program execution (finds control-S)
930 - 946	Input variables for program execution (4 variables)
950 - 975	Input variables for program execution (2 variables; finds control-S)
980 - 988	Input variables for program execution (2 variables)
990 - 998	Input variables for program execution (3 variables)
1000 - 1100	Time and date subroutine
1272 - 1450	Execution of COR command list program
1450 - 1600	Print results subroutine

Table XIX (Continued)

Line Numbers	Comments
1600 - 1700	Call data on disk
2000 - 6004	Plotter subroutine
6100 - 6220	Plotter peripherals
6300 - 7000	Plotter peripherals

Automatic

Automatic operation is achieved by use of the program Master. Comment statements appear on the monitor during execution of this program. These comments inform the user as to the stage of execution.

To operate one must first boot up the Apple (Appendix A). Next type,

RUN MASTER.

The program will then ask several questions relating to data set manipulation and execution. A flashing cursor will eventually appear telling the user when to connect the telephone receiver to the modem. After data transfer and command list program execution, the Apple will collect time and date information, logoff, and finally print out the results on the plotter.

TABLE XX

EXECUTION TIME

Process	Approximate Min. Elapsed
Input of data to Apple	2 min.
LOGON procedure	1 min.
Clearing of data files	35 sec.
Data file sending subroutine	11 min.
Execution of COR	2 min.
Data file receiving subroutine	10 min.
Time and date	15 sec.
Logoff procedure	10 sec.
Plotting subroutine	36 min.

```
]RUN SHIFT.01
******************
PHASE SHIFT PROGRAM
******************

FREQUENCY USED (MHZ) ? 100.1
INPUT # OF PTS. 1000
ENTER LAMP FILENAME TESTO
INPUT DRIVE NUMBER 2
INPUT DECAY FILENAME TEST30
DRIVE NUMBER 2

INTEGRATION CALCULATION

INTEGRATION BEGINS AT 127

LIFETIME IS 2.48916783E-09 SEC

]PR#0
```

```
PR#1
]RUN SHIFT1

FILE NOT FOUND

BREAK IN 52
]

]



RUN SHIFT1,D1
*****************
PHASE SHIFT PROGRAM
*****************

FREQUENCY USED (MHZ) ? 100.1
INPUT # OF PTS. 1000
ENTER LAMP FILENAME TESTO
INPUT DRIVE NUMBER 2
INPUT DECAY FILENAME TEST30
DRIVE NUMBER 2
LAMP=TESTO DECAY=TEST30 MHZ=100.1


 TAU1 = 2.84941594E-09 SEC

 TAU2 = 2.73134759E-09 SEC

 TAU3 = 2.80704127E-09 SEC

 TAU4 = 2.74117695E-09 SEC

 THE LIFETIME IS ... 2.79550364E-09

 ]
```

APPENDIX B

PROGRAMS FOR THE SINGLE PHOTON COUNTING METHOD

Deconvolution Apple II/IBM 370

System Software Manual

Fortran Deconvolution Programs

1. OSU Dataset Programs
   a. OKLA 1
   b. OKLA 2
   c. OKLA 3
   d. PHASE 1
   e. PHASE 2
2. Command List Programs
   a. COR
   b. CCOR
   c. DOUBLE
   d. DPRINT
   e. PHASEP

The System

The interface software was designed with simplicity in mind. The system can be classified into three key parts. That is, the OSU dataset programs, the command list programs, and the Apple II "basic" programs. Each part plays an important role in the manipulation of data.

The Apple programs execute with direct commands which control the IBM 370. These commands are directed to command list programs which then direct control to the OSU dataset programs.

Two types of operation are available to the user. That is, manual or automatic operation are available. The manual operating system is designed for flexibility in the processing of data.

OSU Dataset Programs

The OSU dataset algorithms combine the best features of a gradient search and a method of linearizing the actual function. The program uses a gradient search far away from the solution, and an analytical function as convergence is neared.

The linear function is approximated as,

$$y(x) = y_o(x) + \sum_{j=1}^{n} \left(\frac{\partial\, y_o(x)}{\partial\, a_j} \cdot \delta\, a_j\right) \tag{1}$$

(This fitting function is evaluated with the subroutine FUNCTN: derivatives are evaluated by subroutine DERIV.)

Upon taking a parabolic expansion for a function Chi Squared, one can take its derivative with respect to the independent parameters (a_j).

$$\frac{\partial\, x^2}{\partial\, \delta a_k} = -2\, \Sigma\left(\frac{1}{\sigma_i^2} \left\{y_i - y_o(x_i) - \sum_{j=i}^{n} \frac{\partial\, y_o(x_i)}{\partial\, a_j}\, \delta\, a_j\right\} x\, \frac{\partial\, y_o(x_i)}{\partial\, a_j}\right) \tag{2}$$

One can obtain a set of n simultaneous equations by setting equation (2) to zero.

Where:

$$B_k = \sum_{j=1}^{n} (\delta\, a_j \cdot \alpha_{jk})$$

and,

$$\alpha_{jk} \cong \Sigma\left(\frac{1}{a_i^2} \cdot \frac{\partial\, y_o(x_i)}{\partial\, a_j} \cdot \frac{\partial\, y_o(x_i)}{\partial\, a_k}\right)$$

In order to use the gradient expansion method one incorporates an incrementing parameter (λ) and redefines ∂_{jk} as,

$$\alpha'_{jk} = \begin{cases} \alpha_{jk}\ (1+\lambda) & \text{for } j=k \\ \alpha_{jk} & \text{for } j\neq k \end{cases}$$

The solution of $\delta\ a_j$ follows from a matrix inversion (MATINV SUB-ROUTINE)

$$\delta\ a_j = \sum_{k=1}^{n} (B_k\ \varepsilon'_{jk})$$

The algorithm recipe (from Marquardt) is as follows:

1. Compute $X^2(\dot{a})$.
2. Start initially with $\lambda = 0.001$.
3. Compute δa and $X^2(a+\delta a)$ with this choice of λ.
4. If $X^2\ (a+\delta a) > X^2(a)$, increase λ by a factor of 10 and repeat step (3). (ICOUNT LOOP)
5. If $X^2\ (a+\delta a) < X^2(a)$, decrease λ by a factor of 10, consider $a' = a+\delta a$ to be the new starting point, and return to step (3) substituting a' for a. (JCOUNT LOOP).

OKLA 1

This program calculates the lifetime and returns a value of Chi Square. This program is for manual operation and can be accessed through the card reader.

A description of the accessing program and its simple output is given on the following pages.

```
00000000011111111112222222222333333333344444444445555555555666666666677777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890
CARD
//OKLAHOMA JOB (13530,441-62-4034),'BOB',TIME=(0,40),CLASS=A
//  EXEC FORTGCL
//FORT.SYSIN DD *
      DIMENSION ANK(4),BETA(4)
      DIMENSION ALPHA(4,4)
      DIMENSION NET(1024),LET(1024)
C
      COMMON/AREA1/P(1024)
      COMMON/AREA2/Y(1024),C(1024),WEIGHT(1024)
      COMMON/AREA4/D(4,1024)
      COMMON/AREA5/ITITLE(17)
      COMMON/AREA6/B(4),A(4)
      COMMON/AREA7/ARRAY(4,4)
      COMMON/AREA8/SIGMAA(4)
C
  976 WRITE(6,2)
    2 FORMAT(1X,'READ IN A TITLE:(17A4)')
      READ(5,3)ITITLE
    3 FORMAT(17A4)
      WRITE(6,4)ITITLE
    4 FORMAT(1X,' TITLE : ',17A4)
C
  711 WRITE(6,5)
    5 FORMAT(1X,'HOW MANY CHANNELS ?(I4)')
      READ(5,6)NCHAN
    6 FORMAT(I4)
      WRITE(6,7)NCHAN
    7 FORMAT(1X,'NO. OF CHANNELS= ',I4)
  493 M=M+10
      N=M-9
      READ(1,491)(LET(I),I=1,10)
      READ(2,491)(NET(I),I=1,10)
      I=0
      J=N-1
  498 I=I+1
      J=J+1
      IF(J.GT.NCHAN)GO TO 8
      IF(I.GT.10)GO TO 493
      P(J)=FLOAT(LET(I))
      Y(J)=FLOAT(NET(I))
      GO TO 498
  491 FORMAT(10(1X,I6))
C
C
    8 WRITE(6,14)
   14 FORMAT(1H1,10X,'LAMP DATA')
      NNCHAN=NCHAN/10
      DO 752 J=1,NNCHAN
      L=J*10-9
      LLL=L+9
      WRITE(6,762)(P(K),K=L,LLL)
  752 CONTINUE
      WRITE(6,777)
```

```
                          80/80 LIST                          PAGE  003

000000000111111111122222222223333333333444444444455555555556666666666777777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890
CARD
   67 CALL FUNCT2(N1,N2,FACT,CHISQR)
C
      JCOUNT=0
      FLAMDA=0.001
C
      WRITE(6,34)CHISQR
   34 FORMAT(11X,' CHISQR= ',E14.5)
C
   80 CALL DERIV2(N1,N2,FACT)
C
   90 JCOUNT=JCOUNT+1
C
      DO 91 J=1,4
      BETA(J)=0.0
      DO 91 K=1,J
   91 ALPHA(J,K)=0.0
      DO 110 I=N1,N2
      DO 100 J=1,4
      BETA(J)=BETA(J)+WEIGHT(I)*(Y(I)-C(I))*D(J,I)
      DO 100 K=1,J
  100 ALPHA(J,K)=ALPHA(J,K)+WEIGHT(I)*D(J,I)*D(K,I)
  110 CONTINUE
      DO 150 J=1,4
      DO 150 K=1,J
  150 ALPHA(K,J)=ALPHA(J,K)
C
  151 CHISQ1=CHISQR
      ICOUNT=0
C
  160 DO 174 J=1,4
      DO 173 K=1,4
  173 ARRAY(J,K)=ALPHA(J,K)/SQRT(ALPHA(J,J)*ALPHA(K,K))
  174 ARRAY(J,J)=1.0+FLAMDA
C
      CALL MATINV(4,DET)
C
      ICOUNT=ICOUNT+1
C
      DO 201 J=1,4
      B(J)=A(J)
      DO 201 K=1,4
      B(J)=B(J)+BETA(K)*ARRAY(J,K)/SQRT(ALPHA(J,J)*ALPHA(K,K))
  201 CONTINUE
C
      DO 612 J=1,4
      ANK(J)=A(J)
      A(J)=B(J)
  612 CONTINUE
C
  300 CALL FUNCT2(N1,N2,FACT,CHISQR)
C
      WRITE(6,660)
  660 FORMAT(1X)
      WRITE(6,867)FLAMDA,CHISQR
  867 FORMAT(1X,'FLAMDA=',E15.4,5X,'CHISQR=',E15.4)
```

```
00000000011111111112222222222333333333344444444445555555555666666666677777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890
  777 FORMAT(1H1,10X,'DECAY DATA')
      DO 753 J=1,NNCHAN
      L=J*10-9
      LLL=L+9
      WRITE(6,762)(Y(K),K=L,LLL)
  753 CONTINUE
  762 FORMAT(1X,10(3X,F7.1))
      WRITE(6,19)
   19 FORMAT(1H1,'READ IN LAMP BACKGROUND (F7.3)')
      READ(5,20)BKGDP
      WRITE(6,919)
  919 FORMAT(1X,'READ IN DECAY BACKGROUND (F7.3)')
      READ(5,20)BKGDY
   20 FORMAT(2F7.3)
      WRITE(6,231)BKGDP
      WRITE(6,232)BKGDY
  231 FORMAT(1X,'EXCITATION BACKGROUND=',F7.3)
  232 FORMAT(1X,'DECAY BACKGROUND    =',F7.3)
  703 FACT=0.097
      WRITE(6,23)FACT
   23 FORMAT(1X,'CHANNEL WIDTH    =',F7.3)
   24 Y(1)=BKGDY
      P(I)=BKGDP
      DO 25 I=1,NCHAN
      P(I)=P(I)-BKGDP
      IF(P(I).LT.0.0) P(I)=0.0
      IF(Y(I).LT.1.0) Y(I)=1.0
      WEIGHT(I)=1.0/Y(I)
      Y(I)=Y(I)-BKGDY
      IF(Y(I).LT.0.0) Y(I)=0.0
   25 CONTINUE
C
C
      READ(5,6)N1
      READ(5,6)N2
      WRITE(6,929)N1,N2
  929 FORMAT(1X,'CHANNEL FITTING RANGE IS ',I4,' TO ',I4)
  707 WRITE(6,29)
   29 FORMAT(1X,'ESTIMATE A1 (F5.2)')
      READ(5,30)A(1)
      WRITE(6,690)
  690 FORMAT(1X,'A2')
      READ(5,30)A(3)
      WRITE(6,691)
  691 FORMAT(1X,'TAU 1')
      READ(5,30)A(2)
      WRITE(6,692)
  692 FORMAT(1X,'TAU 2')
      READ(5,30)A(4)
   30 FORMAT(F5.2)
   69 WRITE(6,31)A(1),A(2)
   31 FORMAT(1X,'A1=',E14.5,5X,' TAU1= ',F10.6)
      WRITE(6,32)A(3),A(4)
   32 FORMAT(1X,'A2=',E14.5,5X,' TAU2= ',F10.6)
C
```

```
0000000001111111111222222222233333333334444444444555555555566666666667777777778
1234567890123456789012345678901234567890123456789012345678901234567890123456789 0
      WRITE(6,660)
      DO 619 J=1,4
      A(J)=ANK(J)
  619 CONTINUE
C
C
      IF(B(2).LE.0.0)GO TO 302
      IF(B(4).LE.0.0)GO TO 302
C
      IF(CHISQ1-CHISQR)302,303,303
C
  303 DO 304 J=1,4
      A(J)=B(J)
      SIGMAA(J)=SQRT(ARRAY(J,J)/ALPHA(J,J))
  304 CONTINUE
C
      IF(JCOUNT.EQ.75)GO TO 560
      IF((CHISQ1-CHISQR).LT.0.000001)GO TO 560
C
      FLAMDA=FLAMDA/10.0
      GO TO 80
C
  302 FLAMDA=FLAMDA*10.0
      IF(ICOUNT.EQ.25)GO TO 305
      IF(ICOUNT.EQ.26)GO TO 330
      GO TO 160
C
  305 FLAMDA=FLAMDA/10**6
      GO TO 160
C
  330 CHISQR=CHISQ1
  360 DO 369 J=1,4
      SIGMAA(J)=SQRT(ARRAY(J,J)/ALPHA(J,J))
  369 CONTINUE
C
  560 WRITE(6,708)ICOUNT
      WRITE(6,709)JCOUNT
      WRITE(6,720)
      WRITE(6,53)N1,N2,CHISQR
      WRITE(6,54)A(1),SIGMAA(1)
      WRITE(6,55)A(2),SIGMAA(2)
      WRITE(6,56)A(3),SIGMAA(3)
      WRITE(6,57)A(4),SIGMAA(4)
   53 FORMAT(1X,' FITTING BETWEEN CHANNELS ',I4,' AND ',I4,2X,' CHISQR=
     $',E14.5)
   54 FORMAT(1X,'A1 =',E14.5,2X,'STD.DEV.=',E14.5)
   55 FORMAT(1X,'TAU1=',E14.5,2X,'STD.DEV.=',E14.5)
   56 FORMAT(1X,'A2 =',E14.5,2X,'STD.DEV.=',E14.5)
   57 FORMAT(1X,'TAU2=',E14.5,2X,'STD.DEV.=',E14.5)
  708 FORMAT(1X,'NO. OF ITERATIONS BY ICOUNT',I6)
  709 FORMAT(1X,'NO. OF ITERATIONS BY JCOUNT',I6)
  720 FORMAT(1X,'MAX. NO. ICOUNT EQ 26:JCOUNT EQ 75')
  940 FORMAT(I1)
C
C
```

```
 000000000111111111122222222223333333333444444444455555555556666666666777777777778
 12345678901234567890123456789012345678901234567890123456789012345678901234567890
CARD
      CALL RPLOT(FACT,CHISQR,NCHAN)
C
  427 STOP
      END
      SUBROUTINE FUNCT2(N1,N2,FACT,CHISQR)
      COMMON/AREA1/P(1024)
      COMMON/AREA2/Y(1024),C(1024),WEIGHT(1024)
      COMMON/AREA3/C1(1024),C2(1024)
      COMMON/AREA6/B(4),A(4)
C
      CMULT=FACT*0.5
C
      C1(1)=CMULT*A(1)*P(1)
      C2(1)=CMULT*A(3)*P(1)
      C(1)=C1(1)+C2(1)
      DO 2 I=2,N2
      C1(I)=EXP(-FACT/A(2))*C1(I-1)+A(1)*CMULT*(P(I-1)*EXP(-FACT/A(2))+P
     $(I))
      C2(I)=EXP(-FACT/A(4))*C2(I-1)+A(3)*CMULT*(P(I-1)*EXP(-FACT/A(4))+P
     $(I))
      C(I)=C1(I)+C2(I)
    2 CONTINUE
C
      NPTS=N2-N1+1
      NFREE=NPTS-4
      FREE=FLOAT(NFREE)
      CHISQ=0.0
      IF(NFREE.LE.0)GO TO 4
      DO 3 I=N1,N2
      CHISQ=CHISQ+WEIGHT(I)*(Y(I)-C(I))*(Y(I)-C(I))
    3 CONTINUE
    4 CHISQR=CHISQ/FREE
      RETURN
      END
      SUBROUTINE DERIV2(N1,N2,FACT)
      COMMON/AREA1/P(1024)
      COMMON/AREA3/C1(1024),C2(1024)
      COMMON/AREA4/D(4,1024)
      COMMON/AREA6/B(4),A(4)
C
      DO 1 I=N1,N2
      D(1,I)=C1(I)/A(1)
      D(3,I)=C2(I)/A(3)
    1 CONTINUE
C
      D(2,1)=0.0
      D(4,1)=0.0
      DO 2 I=2,N2
      P12=EXP(-FACT/A(2))*D(2,I-1)
      P22=FACT*EXP(-FACT/A(2))*C1(I-1)/(A(2)*A(2))
      P32=A(1)*FACT*FACT*P(I-1)*EXP(-FACT/A(2))/(A(2)*A(2))
      D(2,I)=P12+P22+P32
      P14=EXP(-FACT/A(4))*D(4,I-1)
      P24=FACT*EXP(-FACT/A(4))*C2(I-1)/(A(4)*A(4))
      P34=A(3)*FACT*FACT*P(I-1)*EXP(-FACT/A(4))/(A(4)*A(4))
```

```
 00000000011111111112222222222333333333344444444445555555555666666666677777777778
 12345678901234567890123456789012345678901234567890123456789012345678901234567890
      D(4,I)=P14+P24+P34
    2 CONTINUE
      RETURN
      END
      SUBROUTINE MATINV(NORDER,DET)
      DIMENSION IK(10),JK(10)
C
      COMMON/AREA7/ARRAY(4,4)
C
      DET=1.0
      DO 100 K=1,NORDER
      AMAX=0.0
   21 DO 30 I=K,NORDER
      DO 30 J=K,NORDER
      IF(ABS(AMAX)-ABS(ARRAY(I,J)))24,24,30
   24 AMAX=ARRAY(I,J)
      IK(K)=I
      JK(K)=J
   30 CONTINUE
      IF(AMAX)41,32,41
   32 DET=0.0
      GO TO 140
   41 I=IK(K)
      IF(I-K)21,51,43
   43 DO 50 J=1,NORDER
      SAVE=ARRAY(K,J)
      ARRAY(K,J)=ARRAY(I,J)
   50 ARRAY(I,J)=-SAVE
   51 J=JK(K)
      IF(J-K)21,61,53
   53 DO 60 I=1,NORDER
      SAVE=ARRAY(I,K)
      ARRAY(I,K)=ARRAY(I,J)
   60 ARRAY(I,J)=-SAVE
   61 DO 70 I=1,NORDER
      IF(I-K)63,70,63
   63 ARRAY(I,K)=-ARRAY(I,K)/AMAX
   70 CONTINUE
      DO 80 I=1,NORDER
      DO 80 J=1,NORDER
      IF(I-K)74,80,74
   74 IF(J-K)75,80,75
   75 ARRAY(I,J)=ARRAY(I,J)+ARRAY(I,K)*ARRAY(K,J)
   80 CONTINUE
      DO 90 J=1,NORDER
      IF(J-K)83,90,83
   83 ARRAY(K,J)=ARRAY(K,J)/AMAX
   90 CONTINUE
      ARRAY(K,K)=1.0/AMAX
  100 DET=DET*AMAX
      DO 130 L=1,NORDER
      K=NORDER-L+1
      J=IK(K)
      IF(J-K)111,111,105
  105 DO 110 I=1,NORDER
```

```
000000000111111111122222222223333333333444444444455555555556666666666777777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890
      SAVE=ARRAY(I,K)
      ARRAY(I,K)=-ARRAY(I,J)
  110 ARRAY(I,J)=SAVE
  111 I=JK(K)
      IF(I-K)130,130,113
  113 DO 120 J=1,NORDER
      SAVE=ARRAY(K,J)
      ARRAY(K,J)=-ARRAY(I,J)
  120 ARRAY(I,J)=SAVE
  130 CONTINUE
  140 RETURN
      END
      SUBROUTINE RPLOT(FACT,CHISQR,NCHAN)
      DIMENSION XARRAY(1026)
      DIMENSION WORKSP(6000)
      DIMENSION IA(4)
C
      INTEGER LIFE(2),IT1(2),ICQ(2),IT2(2)
      INTEGER IBCB(3),IBCD(3)
C
      COMMON/AREA1/P(1024)
      COMMON/AREA2/Y(1024),C(1024),WEIGHT(1024)
      COMMON/AREA6/B(4),A(4)
      COMMON/AREA8/SIGMAA(4)
C
      DATA LIFE(1),LIFE(2)/'LIFE','TIME'/
      DATA IA1,IA2,IT1(1),IT1(2),IT2(1),IT2(2)/'A1 =','A2 =','TAU1',' ='
     $,'TAU2',' ='/
      DATA ICQ(1),ICQ(2)/'CHIS','QR= '/
      DATA IA(1),IA(2),IA(3),IA(4)/'DECO','NVOL','UTIO','N'/
      DATA IBCD(1),IBCD(2),IBCD(3)/'TIME',' NSE','C'/
      DATA IBCB(1),IBCB(2),IBCB(3)/'INTE','NSIT','Y'/
C
      A1=A(1)
      A2=A(3)
      TAU1=A(2)
      TAU2=A(4)
      CHO=CHISQR
C
      ST=0.0
      SS=0.0
      SK=0.0
      SJ=0.0
      FC=0.0
      FD=0.0
C
      XARRAY(1)=0.0
      DO 601 I=1,NCHAN
      IF(P(I).GT.SJ)SJ=P(I)
      IF(SJ.GT.SS)SS=P(I)
      IF(Y(I).GT.SK)SK=Y(I)
      IF(SK.GT.ST)ST=Y(I)
  601 CONTINUE
      IF(SS.LT.ST)FC=ST/SS
      IF(SS.GT.ST)FD=SS/ST
```

80/80 LIST PAGE 008

```
  00000000011111111112222222222333333333344444444445555555555666666666677777777778
  12345678901234567890123456789012345678901234567890123456789012345678901234567890
CARD
        IF(FC.GT.FD)GO TO 90
        DO 701 I=1,NCHAN
        Y(I)=FD*Y(I)
   701  CONTINUE
        GO TO 100
    90  DO 110 I=1,NCHAN
        P(I)=FC*P(I)
   110  CONTINUE
   100  DO 20 I=2,NCHAN
        XARRAY(I)=XARRAY(I-1)+FACT
    20  CONTINUE
  C
        CALL PLOTS(WORKSP,6000,15)
        CALL PLOT(0.0,0.0,-3)
        CALL PLOT(0.0,8.0,2)
        CALL PLOT(11.,8.0,2)
        CALL PLOT(11.,0.0,2)
        CALL PLOT(0.0,0.0,2)
        CALL SCALE(XARRAY,11.,NCHAN,1)
        CALL SCALE(P,7.0,NCHAN,1)
        CALL SCALE(Y,7.0,NCHAN,1)
        CALL LINE(XARRAY,P,NCHAN,1,-1,83)
        CALL PLOT(0.0,0.0,3)
        CALL PLOT(0.0,0.0,2)
        CALL LINE(XARRAY,Y,NCHAN,1,-1,68)
        CALL AXIS(1.0,7.9,IBCD,-9,9.0,0.0,10.0,XARRAY(NCHAN+2))
        CALL SYMBOL(1.0,0.3,0.7,IBCB,90.0,9)
        CALL SYMBOL(1.5,6.8,0.7,IA,0.0,13)
        CALL SYMBOL(8.0,5.8,0.28,LIFE,0.0,8)
        CALL SYMBOL(8.0,5.4,0.28,IA1,0.0,4)
        CALL SYMBOL(8.0,5.0,0.28,IA2,0.0,4)
        CALL SYMBOL(8.0,4.8,0.28,IT1,0.0,6)
        CALL SYMBOL(8.0,4.4,0.28,IT2,0.0,6)
        CALL SYMBOL(8.0,4.0,0.28,ICO,0.0,8)
        CALL NUMBER(9.0,5.4,0.28,A1,0.0,5)
        CALL NUMBER(9.0,5.0,0.28,A2,0.0,5)
        CALL NUMBER(9.0,4.8,0.28,TAU1,0.0,5)
        CALL NUMBER(9.0,4.4,0.28,TAU2,0.0,5)
        CALL NUMBER(9.0,4.0,0.28,CHO,0.0,5)
        CALL PLOT(10.0,0.0,999)
        RETURN
        END
  //LKED.SYSLMOD DD DSNAME=OSU.ACT13530.PROGRAM(OKLA1),DISP=OLD
  //LKED.LIB DD DSN=SYS7.COMPLOT.PRTPLT,DISP=SHR
  //LKED.SYSIN DD *
    INCLUDE LIB(PLOTS)
  //
  $ENDLIST
```

```
list 'osu.act12288.comlist(dprint)'
 'OSU.ACT12288.COMLIST(DPRINT)'
PROC 3 LAMP DECAY PDATA
/**********
 *  DOUBLE EXP PRINT
/**********
WRITE BATCH JOB DOUBLE EXP
WRITE FOR OSU.ACT12288.PROGRAM(OKLA1)
WRITE
WRITE PROCESSING ON &SYSDATE AT &SYSTIME
WRITE
WRITE ENTER THE JOBNAME
READ JOBNAME
WRITE ENTER THE PROJECT NUMBER
READ PROJ
WRITE ENTER THE PASSWORD
READ PWORD
WRITE
WRITE  JOB BUILT AND SUBMITTED
CONTROL NOMSG
DELETE JOBSTREM.CNTL
CONTROL MSG
QED JOBSTREM.CNTL NEW EMODE
010 //&JOBNAME JOB (&PROJ,441-62-4034),BOBN,TIME=(,5),CLASS=F,
020 //  MSGCLASS=A,NOTIFY=&SYSUID
030 ##PASSWORD &PWORD
040 ##ROUTE  PRINT LOCAL
050 //  EXEC PGM=OKLA1
060 //STEPLIB DD DSN=OSU.ACT12288.PROGRAM,DISP=SHR
070 //FT01F001 DD DSN=U12288A.&LAMP..DATA,DISP=SHR
080 //FT02F001 DD DSN=U12288A.&DECAY..DATA,DISP=SHR
090 //FT05F001 DD DSN=U12288A.&PDATA..DATA,DISP=SHR
100 //FT06F001 DD SYSOUT=A
110 //
C 10 99999 '##' '/*' ALL
SAVE
SUBMIT JOBSTREM.CNTL
END NOSAVE
CONTROL NOMSG
DELETE JOBSTREM.CNTL
CONTROL MSG
READY
end
READY
```

Accessing Program for OKLA 1

Data Input

Line 4 - lamp name

5 - decay name

7 - title

8 - number of points

9 - background counts

10 - offset

11 - curve fit start

12 - curve fit end

13 - A1 coefficient

14 - A2 coefficient

15 - Tan 1

16 - Tan 2

OKLA 2

This program calculates the lifetime and returns a value for Chi Square. Accessing is accomplished through a TSO terminal using the command list routine DOUBLE.

```
$JOB LIST                                                                    12:27:14.137  82/10/23  MVS/R3 7C
                                        80/80 LIST                                        PAGE  001

      00000000011111111112222222222333333333344444444445555555555666666666677777777778
      12345678901234567890123456789012345678901234567890123456789012345678901234567890
CARD
0001  //OKLAHOMA JOB (13530,441-62-4034),'BOB',TIME=(0.40),CLASS=A
0002  //     EXEC PGM=OKLA1
0003  //STEPLIB DD DSN=OSU.ACT13530.PROGRAM,DISP=SHR
0004  //FT01F001 DD DSN=U13530B.TESTL.DATA,DISP=SHR
0005  //FT02F001 DD DSN=U13530B.TESTD.DATA,DISP=SHR
0006  //FT05F001 DD *
0007  BYN
0008  1000
0009   0.0
0010    0.0
0011  0450
0012  1000
0013   1.0
0014   1.0
0015   4.0
0016   8.0
0017  //FT06F001 DD SYSOUT=A
0018  //FT15F001 DD SYSOUT=A,DCB=(RECFM=UA,BLKSIZE=133)
0019  //
0020  $ENDLIST
```

```
     00000000011111111112222222222333333333344444444445555555555666666666677777777778
     12345678901234567890123456789012345678901234567890123456789012345678901234567890
CARD
//OKLAHOMA JOB (13530,441-62-4034),'BOB',TIME=(0,40),CLASS=A
//  EXEC FORTGCL
//FORT.SYSIN DD *
      DIMENSION ANK(4),BETA(4),SIGMAA(4)
      DIMENSION ALPHA(4,4)
      DIMENSION NET(1024),LET(1024)
C
      COMMON/AREA1/P(1024)
      COMMON/AREA2/Y(1024),C(1024),WEIGHT(1024)
      COMMON/AREA4/D(4,1024)
      COMMON/AREA5/ITITLE(17)
      COMMON/AREA6/B(4),A(4)
      COMMON/AREA7/ARRAY(4,4)
C
      JYES=1
      IF(JYES.EQ.1)GO TO 711
  493 M=M+10
      N=M-9
      READ(1,491)(LET(I),I=1,10)
      READ(2,491)(NET(I),I=1,10)
      I=0
      J=N-1
  498 I=I+1
      J=J+1
      IF(J.GT.NCHAN)GO TO 8
      IF(I.GT.10)GO TO 493
      P(J)=FLOAT(LET(I))
      Y(J)=FLOAT(NET(I))
      GO TO 498
  491 FORMAT(10(1X,I6))
C
    8 JYES=2
  976 WRITE(6,2)
    2 FORMAT(1X,'READ IN A TITLE:(17A4)')
      READ(5,3)ITITLE
    3 FORMAT(17A4)
      WRITE(6,4)ITITLE
    4 FORMAT(1X,' TITLE : ',17A4)
      IF(JYES.EQ.2)GO TO 859
C
  711 WRITE(6,5)
    5 FORMAT(1X,'HOW MANY CHANNELS ?(I4)')
      READ(5,6)NCHAN
    6 FORMAT(I4)
      WRITE(6,7)NCHAN
    7 FORMAT(1X,'NO. OF CHANNELS= ',I4)
      WRITE(6,814)
  814 FORMAT(1X,'IS NO. OF CHANNELS CORRECT ? ')
      WRITE(6,710)
  710 FORMAT(1X,'IF YES TYPE 1:IF NO TYPE 0')
      READ(5,940)KYES
      IF(KYES.EQ.1)GO TO 859
      GO TO 711
```

```
     00000000011111111112222222222333333333344444444445555555555666666666677777777778
     12345678901234567890123456789012345678901234567890123456789012345678901234567890
C
C
  859 IF(JYES.EQ.1)GO TO 493
      WRITE(6,19)
   19 FORMAT(1X,'READ IN LAMP BACKGROUND(F7.3)')
      READ(5,20)BKGDP
      WRITE(6,919)
  919 FORMAT(1X,'READ IN DECAY BACKGROUND (F7.3)')
      READ(5,20)BKGDY
   20 FORMAT(2F7.3)
      WRITE(6,231)BKGDP
      WRITE(6,232)BKGDY
  231 FORMAT(1X,'EXCITATION BACKGROUND=',F7.3)
  232 FORMAT(1X,'DECAY BACKGROUND     =',F7.3)
      WRITE(6,701)
  701 FORMAT(1X,'IS LAMP AND DECAY BACKGROUND CORRECT ?')
      WRITE(6,702)
  702 FORMAT(1X,'IF YES TYPE 1:IF NO TYPE 0')
      READ(5,940)NYES
      IF(NYES.EQ.1)GO TO 703
      GO TO 8
  703 FACT=0.097
      WRITE(6,23)FACT
   23 FORMAT(1X,'CHANNEL WIDTH     =',F7.3)
   24 Y(I)=BKGDY
      P(I)=BKGDP
      DO 25 I=1,NCHAN
      P(I)=P(I)-BKGDP
      IF(P(I).LT.0.0) P(I)=0.0
      IF(Y(I).LT.1.0) Y(I)=1.0
      WEIGHT(I)=1.0/Y(I)
      Y(I)=Y(I)-BKGDY
      IF(Y(I).LT.0.0) Y(I)=0.0
   25 CONTINUE
C
C
   26 WRITE(6,27)
   27 FORMAT(1X,'CHANNEL FITTING RANGE IS (I4) (HIT RETURN) TO (I4).')
      READ(5,6)N1
      READ(5,6)N2
      WRITE(6,929)N1,N2
  929 FORMAT(1X,'CHANNEL FITTING RANGE IS ',I4,' TO ',I4)
      WRITE(6,921)
  921 FORMAT(1X,'IS CHANNEL FITTING RANGE CORRECT ?')
      WRITE(6,922)
  922 FORMAT(1X,'IF YES TYPE 1:IF NO TYPE 0')
      READ(5,940)MYES
      IF(MYES.EQ.1)GO TO 707
      GO TO 26
  707 WRITE(6,29)
   29 FORMAT(1X,'ESTIMATE A1 (F5.2)')
      READ(5,30)A(1)
      WRITE(6,690)
  690 FORMAT(1X,'A2')
      READ(5,30)A(3)
```

```
000000000111111111122222222223333333333444444444455555555556666666666777777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890
      WRITE(6,691)
  691 FORMAT(1X,'TAU 1')
      READ(5,30)A(2)
      WRITE(6,692)
  692 FORMAT(1X,'TAU 2')
      READ(5,30)A(4)
   30 FORMAT(F5.2)
   69 WRITE(6,31)A(1),A(2)
   31 FORMAT(1X,'A1=',E14.5,5X,' TAU1= ',F10.6)
      WRITE(6,32)A(3),A(4)
   32 FORMAT(1X,'A2=',E14.5,5X,' TAU2= ',F10.6)
      WRITE(6,705)
  705 FORMAT(1X,'ARE FITTING CONSTANTS CORRECT ?')
      WRITE(6,706)
  706 FORMAT(1X,'IF YES TYPE 1:IF NO TYPE 0')
      READ(5,940)LYES
      IF(LYES.EQ.1)GO TO 67
      GO TO 707
C
   67 CALL FUNCT2(N1,N2,FACT,CHISQR)
C
      JCOUNT=0
      FLAMDA=0.001
C
      WRITE(6,34)CHISQR
   34 FORMAT(11X,' CHISQR= ',E14.5)
C
   80 CALL DERIV2(N1,N2,FACT)
C
   90 JCOUNT=JCOUNT+1
C
      DO 91 J=1,4
      BETA(J)=0.0
      DO 91 K=1,J
   91 ALPHA(J,K)=0.0
      DO 110 I=N1,N2
      DO 100 J=1,4
      BETA(J)=BETA(J)+WEIGHT(I)*(Y(I)-C(I))*D(J,I)
      DO 100 K=1,J
  100 ALPHA(J,K)=ALPHA(J,K)+WEIGHT(I)*D(J,I)*D(K,I)
  110 CONTINUE
      DO 150 J=1,4
      DO 150 K=1,J
  150 ALPHA(K,J)=ALPHA(J,K)
C
  151 CHISQ1=CHISQR
      ICOUNT=0
C
  160 DO 174 J=1,4
      DO 173 K=1,4
  173 ARRAY(J,K)=ALPHA(J,K)/SQRT(ALPHA(J,J)*ALPHA(K,K))
  174 ARRAY(J,J)=1.0+FLAMDA
C
      CALL MATINV(4,DET)
C
```

```
         000000000111111111122222222223333333333444444444455555555556666666666777777777778
         12345678901234567890123456789012345678901234567890123456789012345678901234567890
CARD
      ICOUNT=ICOUNT+1
C
      DO 201 J=1,4
      B(J)=A(J)
      DO 201 K=1,4
      B(J)=B(J)+BETA(K)*ARRAY(J,K)/SQRT(ALPHA(J,J)*ALPHA(K,K))
  201 CONTINUE
C
      DO 612 J=1,4
      ANK(J)=A(J)
      A(J)=B(J)
  612 CONTINUE
C
  300 CALL FUNCT2(N1,N2,FACT,CHISQR)
C
      DO 619 J=1,4
      A(J)=ANK(J)
  619 CONTINUE
C
C
      IF(B(2).LE.0.0)GO TO 302
      IF(B(4).LE.0.0)GO TO 302
C
      IF(CHISQ1-CHISQR)302,303,303
C
  303 DO 304 J=1,4
      A(J)=B(J)
      SIGMAA(J)=SQRT(ARRAY(J,J)/ALPHA(J,J))
  304 CONTINUE
C
      IF(JCOUNT.EQ.75)GO TO 560
      IF((CHISQ1-CHISQR).LT.0.000001)GO TO 560
C
      FLAMDA=FLAMDA/10.0
      GO TO 80
C
  302 FLAMDA=FLAMDA*10.0
      IF(ICOUNT.EQ.25)GO TO 305
      IF(ICOUNT.EQ.26)GO TO 330
      GO TO 160
C
  305 FLAMDA=FLAMDA/10**6
      GO TO 160
C
  330 CHISQR=CHISQ1
  360 DO 369 J=1,4
      SIGMAA(J)=SQRT(ARRAY(J,J)/ALPHA(J,J))
  369 CONTINUE
C
  560 WRITE(6,708)ICOUNT
      WRITE(6,709)JCOUNT
      WRITE(6,720)
      WRITE(6,53)N1,N2,CHISQR
      WRITE(6,54)A(1),SIGMAA(1)
      WRITE(6,55)A(2),SIGMAA(2)
```

```
         0000000001111111111222222222233333333334444444444555555555566666666667777777777778
         12345678901234567890123456789012345678901234567890123456789012345678901234567890
CARD
          WRITE(6,56)A(3),SIGMAA(3)
          WRITE(6,57)A(4),SIGMAA(4)
       53 FORMAT(1X,' FITTING BETWEEN CHANNELS ',I4,' AND ',I4,2X,' CHISQR=
         $',E14.5)
       54 FORMAT(1X,'A1 =',E14.5,2X,'STD.DEV.=',E14.5)
       55 FORMAT(1X,'TAU1=',E14.5,2X,'STD.DEV.=',E14.5)
       56 FORMAT(1X,'A2 =',E14.5,2X,'STD.DEV.=',E14.5)
       57 FORMAT(1X,'TAU2=',E14.5,2X,'STD.DEV.=',E14.5)
      708 FORMAT(1X,'NO. OF ITERATIONS BY ICOUNT',I6)
      709 FORMAT(1X,'NO. OF ITERATIONS BY JCOUNT',I6)
      720 FORMAT(1X,'MAX. NO. ICOUNT EQ 26:JCOUNT EQ 75')
      940 FORMAT(I1)
C
          WRITE(6,936)
      936 FORMAT(1X,'TO RUN PROGRAM AGAIN TYPE 0')
          WRITE(6,937)
      937 FORMAT(1X,'TO END PROGRAM TYPE 1')
          READ(5,940)JYES
          IF(JYES.EQ.0)GO TO 976
C
C
      427 STOP
          END
          SUBROUTINE FUNCT2(N1,N2,FACT,CHISQR)
          COMMON/AREA1/P(1024)
          COMMON/AREA2/Y(1024),C(1024),WEIGHT(1024)
          COMMON/AREA3/C1(1024),C2(1024)
          COMMON/AREA6/B(4),A(4)
C
          CMULT=FACT*0.5
C
          C1(1)=CMULT*A(1)*P(1)
          C2(1)=CMULT*A(3)*P(1)
          C(1)=C1(1)+C2(1)
          DO 2 I=2,N2
          C1(I)=EXP(-FACT/A(2))*C1(I-1)+A(1)*CMULT*(P(I-1)*EXP(-FACT/A(2))+P
         $(I))
          C2(I)=EXP(-FACT/A(4))*C2(I-1)+A(3)*CMULT*(P(I-1)*EXP(-FACT/A(4))+P
         $(I))
          C(I)=C1(I)+C2(I)
        2 CONTINUE
C
          NPTS=N2-N1+1
          NFREE=NPTS-4
          FREE=FLOAT(NFREE)
          CHISQ=0.0
          IF(NFREE.LE.0)GO TO 4
          DO 3 I=N1,N2
          CHISQ=CHISQ+WEIGHT(I)*(Y(I)-C(I))*(Y(I)-C(I))
        3 CONTINUE
        4 CHISQR=CHISQ/FREE
          RETURN
          END
          SUBROUTINE DERIV2(N1,N2,FACT)
          COMMON/AREA1/P(1024)
```

```
000000000111111111122222222223333333333444444444455555555556666666666777777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890
      COMMON/AREA3/C1(1024),C2(1024)
      COMMON/AREA4/D(4,1024)
      COMMON/AREA6/B(4),A(4)
C
      DO 1 I=N1,N2
      D(1,I)=C1(I)/A(1)
      D(3,I)=C2(I)/A(3)
    1 CONTINUE
C
      D(2,1)=0.0
      D(4,1)=0.0
      DO 2 I=2,N2
      P12=EXP(-FACT/A(2))*D(2,I-1)
      P22=FACT*EXP(-FACT/A(2))*C1(I-1)/(A(2)*A(2))
      P32=A(1)*FACT*FACT*P(I-1)*EXP(-FACT/A(2))/(A(2)*A(2))
      D(2,I)=P12+P22+P32
      P14=EXP(-FACT/A(4))*D(4,I-1)
      P24=FACT*EXP(-FACT/A(4))*C2(I-1)/(A(4)*A(4))
      P34=A(3)*FACT*FACT*P(I-1)*EXP(-FACT/A(4))/(A(4)*A(4))
      D(4,I)=P14+P24+P34
    2 CONTINUE
      RETURN
      END
      SUBROUTINE MATINV(NORDER,DET)
      DIMENSION IK(10),JK(10)
C
      COMMON/AREA7/ARRAY(4,4)
C
      DET=1.0
      DO 100 K=1,NORDER
      AMAX=0.0
   21 DO 30 I=K,NORDER
      DO 30 J=K,NORDER
      IF(ABS(AMAX)-ABS(ARRAY(I,J)))24,24,30
   24 AMAX=ARRAY(I,J)
      IK(K)=I
      JK(K)=J
   30 CONTINUE
      IF(AMAX)41,32,41
   32 DET=0.0
      GO TO 140
   41 I=IK(K)
      IF(I-K)21,51,43
   43 DO 50 J=1,NORDER
      SAVE=ARRAY(K,J)
      ARRAY(K,J)=ARRAY(I,J)
   50 ARRAY(I,J)=-SAVE
   51 J=JK(K)
      IF(J-K)21,61,53
   53 DO 60 I=1,NORDER
      SAVE=ARRAY(I,K)
      ARRAY(I,K)=ARRAY(I,J)
   60 ARRAY(I,J)= SAVE
   61 DO 70 I=1,NORDER
      IF(I-K)63,70,63
```

```
                               80/80 LIST

0000000001111111111222222222233333333334444444444555555555566666666667777777777
1234567890123456789012345678901234567890123456789012345678901234567890123456789O
   63 ARRAY(I,K)=-ARRAY(I,K)/AMAX
   70 CONTINUE
      DO 80 I=1,NORDER
      DO 80 J=1,NORDER
      IF(I-K)74,80,74
   74 IF(J-K)75,80,75
   75 ARRAY(I,J)=ARRAY(I,J)+ARRAY(I,K)*ARRAY(K,J)
   80 CONTINUE
      DO 90 J=1,NORDER
      IF(J-K)83,90,83
   83 ARRAY(K,J)=ARRAY(K,J)/AMAX
   90 CONTINUE
      ARRAY(K,K)=1.0/AMAX
  100 DET=DET*AMAX
      DO 130 L=1,NORDER
      K=NORDER-L+1
      J=IK(K)
      IF(J-K)111,111,105
  105 DO 110 I=1,NORDER
      SAVE=ARRAY(I,K)
      ARRAY(I,K)=-ARRAY(I,J)
  110 ARRAY(I,J)=SAVE
  111 I=JK(K)
      IF(I-K)130,130,113
  113 DO 120 J=1,NORDER
      SAVE=ARRAY(K,J)
      ARRAY(K,J)=-ARRAY(I,J)
  120 ARRAY(I,J)=SAVE
  130 CONTINUE
  140 RETURN
      END
//LKED.SYSLMOD DD DSN=OSU.ACT13530.PROGRAM,
//      DISP=(NEW,CATLG),SPACE=(TRK,(20,5,2)),
//      UNIT=3350,VOL=SER=DASDGO,
//      DCB=(BLKSIZE=19069,RECFM=U,DSORG=PO)
//LKED.SYSIN DD *
  NAME OKLA2(R)
//
//
//
//
$ENDLIST
```

OKLA 3

OKLA 3 calculates the lifetime, a residual file, an autocorrelation file, and a value for Chi Squared.

This program can be accessed by either command list programs CCOR or COR.

Program for Phase 1

This program calculates a single experimental lifetime from the input of a lamp and a decay file.

Program for Phase 2

Program Phase 2 is an interactive version of Phase 1. The program will calculate a lifetime given experimental data using the phase-plane method.

Accessing Program Phase 1

This is the job control language that will access program Phase 1.

Command List Programs

Command procedures are used to allocate datasets and work space for the OSU dataset programs. The purpose of these command programs is to enable the user to access and execute a program in TSO interactive mode with one only one executable statement.

The work space in these command procedures have been allocated in a most efficient manner. The user is not required to estimate the work space needed. Note that all datasets use a "fixed block" record length.

All command list programs can be executed in the READY mode by using an EXEC statement.

Command list programs are only reserved on the disk for 14 days, if not executed. After this, the programs will be reclassified as "OLD". Old datasets can be reclaimed by the procedure listed in III (2).

CCOR

This command list program is primarily intended for manual operation. Its purpose is to allocate data files for OSU dataset program OKLA 3. When executed the program asks for a lamp and decay file name. It then asks for file names for the respective residual and autocorrelation files. Work space is automatically allocated for these four files.

COR

This program is primarily designed for automatic operation. The Apple program "Master" uses an EXEC statement to access this command program. This program will in turn access OSU dataset program OKLA 3. The major difference between COR and CCOR is that in COR the dataset names are fixed. (e.g., lamp-lamp data; CDATA-correlation data; decay-decay data; RES-residual data.) The obvious purpose of fixed dataset names is to prevent a novice operator from being able to collect a large amount of data on the disk without being able to delete it.

12:32:37.266 82/10/23 MVS/R3.7C

80/80 LIST PAGE 001

```
000000000111111111122222222223333333333444444444455555555556666666666777777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890
.//OKLAHOMA JOB (13530,441-62-4034),'BOB',TIME=(0,40),CLASS=A
//   EXEC FORTGCL
//FORT.SYSIN DD *
      DIMENSION ANK(4),BETA(4),SIGMAA(4)
      DIMENSION ALPHA(4,4)
      DIMENSION NET(1024),LET(1024)
C
      COMMON/AREA1/P(1024)
      COMMON/AREA2/Y(1024),C(1024),WEIGHT(1024)
      COMMON/AREA4/D(4,1024)
      COMMON/AREA5/ITITLE(17)
      COMMON/AREA6/B(4),A(4)
      COMMON/AREA7/ARRAY(4,4)
      COMMON/AREA8/SCOR(1024)
C
      JYES=1
      IF(JYES.EQ.1)GO TO 711
  493 M=M+10
      N=M-9
      READ(1,491)(LET(I),I=1,10)
      READ(2,491)(NET(I),I=1,10)
      I=0
      J=N-1
  498 I=I+1
      J=J+1
      IF(J.GT.NCHAN)GO TO 8
      IF(I.GT.10)GO TO 493
      P(J)=FLOAT(LET(I))
      Y(J)=FLOAT(NET(I))
      GO TO 498
  491 FORMAT(10(1X,I6))
C
C
    8 JYES=2
  976 WRITE(6,2)
    2 FORMAT(1X,'READ IN A TITLE:(17A4)')
      READ(5,3)ITITLE
    3 FORMAT(17A4)
      WRITE(6,4)ITITLE
    4 FORMAT(1X,' TITLE : ',17A4)
      IF(JYES.EQ.2)GO TO 859
C
  711 WRITE(6,5)
    5 FORMAT(1X,'HOW MANY CHANNELS ?(I4)')
      READ(5,6)NCHAN
    6 FORMAT(I4)
      WRITE(6,7)NCHAN
    7 FORMAT(1X,'NO. OF CHANNELS= ',I4)
      WRITE(6,814)
  814 FORMAT(1X,'IS NO. OF CHANNELS CORRECT ? ')
      WRITE(6,710)
  710 FORMAT(1X,'IF YES TYPE 1:IF NO TYPE 0')
      READ(5,940)KYES
```

```
 00000000011111111112222222222333333333344444444445555555555666666666677777777778
 12345678901234567890123456789012345678901234567890123456789012345678901234567890
      IF(KYES.EQ.1)GO TO 859
      GO TO 711
C
  859 IF(JYES.EQ.1)GO TO 493
      WRITE(6,19)
   19 FORMAT(1X,'READ IN LAMP BACKGROUND(F7.3)')
      READ(5,20)BKGDP
      WRITE(6,919)
  919 FORMAT(1X,'READ IN DECAY BACKGROUND (F7.3)')
      READ(5,20)BKGDY
   20 FORMAT(2F7.3)
      WRITE(6,231)BKGDP
      WRITE(6,232)BKGDY
  231 FORMAT(1X,'EXCITATION BACKGROUND=',F7.3)
  232 FORMAT(1X,'DECAY BACKGROUND    =',F7.3)
      WRITE(6,701)
  701 FORMAT(1X,'IS LAMP AND DECAY BACKGROUND CORRECT ?')
      WRITE(6,702)
  702 FORMAT(1X,'IF YES TYPE 1:IF NO TYPE 0')
      READ(5,940)NYES
      IF(NYES.EQ.1)GO TO 703
      GO TO 8
  703 FACT=0.097
      WRITE(6,23)FACT
   23 FORMAT(1X,'CHANNEL WIDTH    =',F7.3)
   24 Y(I)=BKGDY
      P(I)=BKGDP
      DO 25 I=1,NCHAN
      P(I)=P(I)-BKGDP
      SCOR(I)=0.0
      IF(P(I).LT.0.0) P(I)=0.0
      IF(Y(I).LT.1.0) SCOR(I)=1.0
      IF(Y(I).LT.1.0) Y(I)=1.0
      WEIGHT(I)=1.0/Y(I)
      Y(I)=Y(I)-BKGDY
      IF(Y(I).LT.0.0) Y(I)=0.0
   25 CONTINUE
C
C
   26 WRITE(6,27)
   27 FORMAT(1X,'CHANNEL FITTING RANGE IS (I4) (HIT RETURN) TO (I4).')
      READ(5,6)N1
      READ(5,6)N2
      WRITE(6,929)N1,N2
  929 FORMAT(1X,'CHANNEL FITTING RANGE IS ',I4,' TO ',I4)
      WRITE(6,921)
  921 FORMAT(1X,'IS CHANNEL FITTING RANGE CORRECT ?')
      WRITE(6,922)
  922 FORMAT(1X,'IF YES TYPE 1:IF NO TYPE 0')
      READ(5,940)MYES
      IF(MYES.EQ.1)GO TO 707
      GO TO 26
  707 WRITE(6,29)
   29 FORMAT(1X,'ESTIMATE A1 (F5.2)')
      READ(5,30)A(1)
```

```
0000000001111111111222222222233333333334444444444555555555566666666667777777778
1234567890123456789012345678901234567890123456789012345678901234567890123456789 0
      WRITE(6,690)
  690 FORMAT(1X,'A2')
      READ(5,30)A(3)
      WRITE(6,691)
  691 FORMAT(1X,'TAU 1')
      READ(5,30)A(2)
      WRITE(6,692)
  692 FORMAT(1X,'TAU 2')
      READ(5,30)A(4)
   30 FORMAT(F5.2)
   69 WRITE(6,31)A(1),A(2)
   31 FORMAT(1X,'A1=',E14.5,5X,' TAU1= ',F10.6)
      WRITE(6,32)A(3),A(4)
   32 FORMAT(1X,'A2=',E14.5,5X,' TAU2= ',F10.6)
      WRITE(6,705)
  705 FORMAT(1X,'ARE FITTING CONSTANTS CORRECT ?')
      WRITE(6,706)
  706 FORMAT(1X,'IF YES TYPE 1:IF NO TYPE 0')
      READ(5,940)LYSS
      IF(LYES.EQ.1)GO TO 67
      GO TO 707
C
   67 CALL FUNCT2(N1,N2,FACT,CHISQR)
C
      JCOUNT=0
      FLAMDA=0.001
C
      WRITE(6,34)CHISQR
   34 FORMAT(11X,' CHISQR= ',E14.5)
C
   80 CALL DERIV2(N1,N2,FACT)
C
   90 JCOUNT=JCOUNT+1
C
      DO 91 J=1,4
      BETA(J)=0.0
      DO 91 K=1,J
   91 ALPHA(J,K)=0.0
      DO 110 I=N1,N2
      DO 100 J=1,4
      BETA(J)=BETA(J)+WEIGHT(I)*(Y(I)-C(I))*D(J,I)
      DO 100 K=1,J
  100 ALPHA(J,K)=ALPHA(J,K)+WEIGHT(I)*D(J,I)*D(K,I)
  110 CONTINUE
      DO 150 J=1,4
      DO 150 K=1,J
  150 ALPHA(K,J)=ALPHA(J,K)
C
  151 CHISQ1=CHISQR
      ICOUNT=0
C
  160 DO 174 J=1,4
      DO 173 K=1,4
  173 ARRAY(J,K)=ALPHA(J,K)/SQRT(ALPHA(J,J)*ALPHA(K,K))
  174 ARRAY(J,J)=1.0+FLAMDA
```

```
     00000000011111111112222222222333333333344444444445555555555666666666677777777778
     12345678901234567890123456789012345678901234567890123456789012345678901234567890
CARD
     C
           CALL MATINV(4,DET)
     C
           ICOUNT=ICOUNT+1
     C
           DO 201 J=1,4
           B(J)=A(J)
           DO 201 K=1,4
           B(J)=B(J)+BETA(K)*ARRAY(J,K)/SQRT(ALPHA(J,J)*ALPHA(K,K))
       201 CONTINUE
     C
           DO 612 J=1,4
           ANK(J)=A(J)
           A(J)=B(J)
       612 CONTINUE
     C
       300 CALL FUNCT2(N1,N2,FACT,CHISQR)
     C
           DO 619 J=1,4
           A(J)=ANK(J)
       619 CONTINUE
     C
     C
           IF(B(2).LE.0.0)GO TO 302
           IF(B(4).LE.0.0)GO TO 302
     C
           IF(CHISQ1-CHISQR)302,303,303
     C
       303 DO 304 J=1,4
           A(J)-B(J)
           SIGMAA(J)=SQRT(ARRAY(J,J)/ALPHA(J,J))
       304 CONTINUE
     C
           IF(JCOUNT.EQ.75)GO TO 560
           IF((CHISQ1-CHISQR).LT.0.000001)GO TO 560
     C
           FLAMDA=FLAMDA/10.0
           GO TO 80
     C
       302 FLAMDA=FLAMDA*10.0
           IF(ICOUNT.EQ.25)GO TO 305
           IF(ICOUNT.EQ.26)GO TO 330
           GO TO 160
     C
       305 FLAMDA=FLAMDA/10**6
           GO TO 160
     C
       330 CHISQR=CHISQ1
       360 DO 369 J=1,4
           SIGMAA(J)=SQRT(ARRAY(J,J)/ALPHA(J,J))
       369 CONTINUE
     C
       560 WRITE(6,708)ICOUNT
           WRITE(6,709)JCOUNT
           WRITE(6,720)
```

```
         000000000111111111122222222223333333333444444444455555555556666666666777777777778
         12345678901234567890123456789012345678901234567890123456789012345678901234567890
      WRITE(6,53)N1,N2,CHISQR
      WRITE(6,54)A(1),SIGMAA(1)
      WRITE(6,55)A(2),SIGMAA(2)
      WRITE(6,56)A(3),SIGMAA(3)
      WRITE(6,57)A(4),SIGMAA(4)
   53 FORMAT(1X,' FITTING BETWEEN CHANNELS ',I4,' AND ',I4,2X,' CHISQR=
     $',E14.5)
   54 FORMAT(1X,'A1 =',E14.5,2X,'STD.DEV.=',E14.5)
   55 FORMAT(1X,'TAU1=',E14.5,2X,'STD.DEV.=',E14.5)
   56 FORMAT(1X,'A2 =',E14.5,2X,'STD.DEV.=',E14.5)
   57 FORMAT(1X,'TAU2=',E14.5,2X,'STD.DEV.=',E14.5)
  708 FORMAT(1X,'NO. OF ITERATIONS BY ICOUNT',I6)
  709 FORMAT(1X,'NO. OF ITERATIONS BY JCOUNT',I6)
  720 FORMAT(1X,'MAX. NO. ICOUNT EQ 26:JCOUNT EQ 75')
  940 FORMAT(I1)
      CALL CORR(NCHAN)
C
      WRITE(6,936)
  936 FORMAT(1X,'TO RUN PROGRAM AGAIN TYPE 0')
      WRITE(6,937)
  937 FORMAT(1X,'TO END PROGRAM TYPE 1')
      READ(5,940)JYES
      IF(JYES.EQ.0)GO TO 976
C
C
  427 STOP
      END
      SUBROUTINE FUNCT2(N1,N2,FACT,CHISQR)
      COMMON/AREA1/P(1024)
      COMMON/AREA2/Y(1024),C(1024),WEIGHT(1024)
      COMMON/AREA3/C1(1024),C2(1024)
      COMMON/AREA6/B(4),A(4)
C
      CMULT=FACT*0.5
C
      C1(1)=CMULT*A(1)*P(1)
      C2(1)=CMULT*A(3)*P(1)
      C(1)=C1(1)+C2(1)
      DO 2 I=2,N2
      C1(I)=EXP(-FACT/A(2))*C1(I-1)+A(1)*CMULT*(P(I-1)*EXP(-FACT/A(2))+P
     $(I))
      C2(I)=EXP(-FACT/A(4))*C2(I-1)+A(3)*CMULT*(P(I-1)*EXP(-FACT/A(4))+P
     $(I))
      C(I)=C1(I)+C2(I)
    2 CONTINUE
C
      NPTS=N2-N1+1
      NFREE=NPTS-4
      FREE=FLOAT(NFREE)
      CHISQ=0.0
      IF(NFREE.LE.0)GO TO 4
      DO 3 I=N1,N2
      CHISQ=CHISQ+WEIGHT(I)*(Y(I)-C(I))*(Y(I)-C(I))
    3 CONTINUE
    4 CHISQR=CHISQ/FREE
```

```
000000000111111111122222222223333333333444444444455555555556666666666777777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890
      RETURN
      END
      SUBROUTINE DERIV2(N1,N2,FACT)
      COMMON/AREA1/P(1024)
      COMMON/AREA3/C1(1024),C2(1024)
      COMMON/AREA4/D(4,1024)
      COMMON/AREA6/B(4),A(4)
C
      DO 1 I=N1,N2
      D(1,I)=C1(I)/A(1)
      D(3,I)=C2(I)/A(3)
    1 CONTINUE
C
      D(2,1)=0.0
      D(4,1)=0.0
      DO 2 I=2,N2
      P12=EXP(-FACT/A(2))*D(2,I-1)
      P22=FACT*EXP(-FACT/A(2))*C1(I-1)/(A(2)*A(2))
      P32=A(1)*FACT*FACT*P(I-1)*EXP(-FACT/A(2))/(A(2)*A(2))
      D(2,I)=P12+P22+P32
      P14=EXP(-FACT/A(4))*D(4,I-1)
      P24=FACT*EXP(-FACT/A(4))*C2(I-1)/(A(4)*A(4))
      P34=A(3)*FACT*FACT*P(I-1)*EXP(-FACT/A(4))/(A(4)*A(4))
      D(4,I)=P14+P24+P34
    2 CONTINUE
      RETURN
      END
      SUBROUTINE MATINV(NORDER,DET)
      DIMENSION IK(10),JK(10)
C
      COMMON/AREA7/ARRAY(4,4)
C
      DET=1.0
      DO 100 K=1,NORDER
      AMAX=0.0
   21 DO 30 I=K,NORDER
      DO 30 J=K,NORDER
      IF(ABS(AMAX)-ABS(ARRAY(I,J)))24,24,30
   24 AMAX=ARRAY(I,J)
      IK(K)=I
      JK(K)=J
   30 CONTINUE
      IF(AMAX)41,32,41
   32 DET=0.0
      GO TO 140
   41 I=IK(K)
      IF(I-K)21,51,43
   43 DO 50 J=1,NORDER
      SAVE=ARRAY(K,J)
      ARRAY(K,J)=ARRAY(I,J)
   50 ARRAY(I,J)=-SAVE
   51 J=JK(K)
      IF(J-K)21,61,53
   53 DO 60 I=1,NORDER
      SAVE=ARRAY(I,K)
```

```
00000000011111111112222222222333333333344444444445555555555666666666677777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890
      ARRAY(I,K)=ARRAY(I,J)
   60 ARRAY(I,J)=-SAVE
   61 DO 70 I=1,NORDER
      IF(I-K)63,70,63
   63 ARRAY(I,K)=-ARRAY(I,K)/AMAX
   70 CONTINUE
      DO 80 I=1,NORDER
      DO 80 J=1,NORDER
      IF(I-K)74,80,74
   74 IF(J-K)75,80,75
   75 ARRAY(I,J)=ARRAY(I,J)+ARRAY(I,K)*ARRAY(K,J)
   80 CONTINUE
      DO 90 J=1,NORDER
      IF(J-K)83,90,83
   83 ARRAY(K,J)=ARRAY(K,J)/AMAX
   90 CONTINUE
      ARRAY(K,K)=1.0/AMAX
  100 DET=DET*AMAX
      DO 130 L=1,NORDER
      K=NORDER-L+1
      J=IK(K)
      IF(J-K)111,111,105
  105 DO 110 I=1,NORDER
      SAVE=ARRAY(I,K)
      ARRAY(I,K)=-ARRAY(I,J)
  110 ARRAY(I,J)=SAVE
  111 I=JK(K)
      IF(I-K)130,130,113
  113 DO 120 J=1,NORDER
      SAVE=ARRAY(K,J)
      ARRAY(K,J)=-ARRAY(I,J)
  120 ARRAY(I,J)=SAVE
  130 CONTINUE
  140 RETURN
      END
      SUBROUTINE CORR(NCHAN)
C
      DIMENSION CR(1024),DEL(1024),DV(1024)
      DIMENSION S(1024)
C
      COMMON/AREA2/Y(1024),C(1024),WEIGHT(1024)
      COMMON/AREA8/SCOR(1024)
      DO 600 I=1,NCHAN
      S(I)=Y(I)
      IF(SCOR(I).GT.0.0)S(I)=0.0
      DEL(I)=C(I)-S(I)
  600 CONTINUE
      DD=0.0
      DO 990 J=1,NCHAN
      DD=(1/Y(J))+DD
  990 CONTINUE
      DD=DD/NCHAN
      DE=0.0
      DO 700 K=1,NCHAN
      DV(K)=(1/Y(K))/DD
```

```
        00000000011111111112222222222333333333344444444445555555555666666666677777777778
        12345678901234567890123456789012345678901234567890123456789012345678901234567890
CARD
              DE=DE+(((DEL(K))**2)*DV(K))
              DV(K)=SQRT(DV(K))
          700 CONTINUE
              DE=DE/NCHAN
              KN=IFIX((NCHAN/2)+0.5)
              DO 750 L=1,KN
              CR(L)=0.0
              DO 850 N=1,KN
              CR(L)=CR(L)+DV(N)*DEL(N)*DV(L+N)*DEL(L+N)
          850 CONTINUE
              CR(L)=CR(L)/(KN*DE)
          750 CONTINUE
          652 FORMAT(1X,10F7.1)
          653 FORMAT(1X,10F7.2)
              I=-9
              N=0
          764 I=I+10
              N=N+10
              IF(I.GT.NCHAN)GO TO 951
              WRITE(3,652)(DEL(L),L=I,N)
              GO TO 764
          951 I=-9
              N=0
          952 I=I+10
              N=N+10
              IF(I.GT.KN)GO TO 955
              WRITE(4,653)(CR(L),L=I,N)
              GO TO 952
          955 RETURN
              END
        //LKED.SYSLMOD DD DSNAME=OSU.ACT13530.PROGRAM,DISP=OLD
        //LKED.SYSIN DD *
          NAME OKLA3(R)
        //
        $ENDLIST
```

```
     00000000011111111112222222222333333333344444444445555555555666666666677777777778
     12345678901234567890123456789012345678901234567890123456789012345678901234567890
CARD
     //OKLAHOMA JOB (13530,441-62-4034),'BOB',TIME=(0,40),CLASS=A
     //  EXEC FORTGCL
     //FORT.SYSIN DD *
     C PHASE-PLANE METHOD
     C  USING LEAST SQUARES
     C
     C
     C
     C WHERE M IS THE INITIAL POINT FOR LEAST SQUARES
     C AND TSPAN IS THE INSTANTANEOUS TIME DIFFERENTIAL
     C AND
     C
     C Z(K)=DSUM/FSUM=FS(K)/DS(K)
     C
     C W(K)=D(K)/DSUM=D(K)/DS(K)
     C
     C AND WHERE,
     C
     C Z(T)=(-TAU*W(K)+K*TAU)
     C
     C K IS EQUAL TO HIT
     C
     C
           DIMENSION FS(1024),DS(1024)
           DIMENSION Z(1024),W(1024)
           DIMENSION X(101),Y(101)
           DIMENSION R(1024),C(1024,3)
           DIMENSION BPAR(4)
           DIMENSION D(1024),F(1024)
     C
     C INITIALIZE VARIABLES
           IC=1024
           BPAR(1)=0.0
           BPAR(2)=0.0
           BPAR(3)=0.0
           BPAR(4)=0.0
           TRES=0.0
           I=0
           SS=0.0
           SUMWW=0.
           SUMWZ=0.
           SUMZ=0.
           SUMW=0.
           SUMZZ=0.0
           XD=0.0
           XL=0.0
           TAS=1.96
     C
     C INITIALIZE OUTPUT ARRAY
     C
     C
     C
     C
```

```
      00000000011111111112222222222333333333344444444445555555555666666666677777777778
      12345678901234567890123456789012345678901234567890123456789012345678901234567890
CARD
            TSPAN=0.097
            R(1)=TSPAN
            A=TSPAN
      C
      C READ VARIABLES
            READ(5,300)N
        300 FORMAT(I4)
            WRITE(6,14)
         14 FORMAT(1H1,10X,'LAMP DATA')
            NNCHAN=N/10
            DO 752 J=1,NNCHAN
            L=J*10-9
            LLL=L+9
            READ(1,11)(F(K),K=L,LLL)
            WRITE(6,762)(F(K),K=L,LLL)
        752 CONTINUE
            WRITE(6,777)
        777 FORMAT(1H1,10X,'DECAY DATA')
            DO 753 J=1,NNCHAN
            L=J*10-9
            LLL=L+9
            READ(2,11)(D(K),K=L,LLL)
            WRITE(6,762)(D(K),K=L,LLL)
            DO 616 II=L,LLL
            IF(SS.LT.D(II))M=II
            IF(SS.LT.D(II))SS=D(II)
        616 CONTINUE
        753 CONTINUE
        762 FORMAT(1X,10(3X,F7.1))
         11 FORMAT(10(1X,F6.0))
            WRITE(6,45)
         45 FORMAT(1H1,'PHASE-PLANE METHOD USING LEAST SQUARES ')
      C
      C
      C  NORMALIZE DECAY CURVES
      C
            DO 660 J=1,N
            IF(F(J).GT.XL)XL=F(J)
            IF(D(J).GT.XD)XD=D(J)
        660 CONTINUE
            DO 665 J=1,N
            D(J)=D(J)*XL/XD
        665 CONTINUE
      C
            DO 22 K=2,N
            R(K)=R(K-1)+TSPAN
         22 CONTINUE
            B=R(N)
      C
      C CALCULATE SUMS
            CALL ICSICU(R,F,N,BPAR,C,IC,IER)
            CALL DCSQDU(R,F,N,C,IC,A,B,Q,IER,FS)
            CALL ICSICU(R,D,N,BPAR,C,IC,IER)
            CALL DCSQDU(R,D,N,C,IC,A,B,Q,IER,DS)
            DO 21 K=1,N
```

```
0000000001111111111222222222233333333334444444444555555555566666666667777777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890

      IF(FS(K).LE.0.0)Z(K)=0.0
      IF(FS(K).LE.0.0)W(K)=0.0
      IF(FS(K).LE.0.0)GO TO 21
   15 Z(K)=DS(K)/FS(K)
      W(K)=D(K)/FS(K)
   21 CONTINUE
C
C
C  DO LEAST SQUARES
C
      N=N-1
      DO 30 K=M,N
      SUMW=SUMW+W(K)
      SUMZ=SUMZ+Z(K)
      SUMWZ=SUMWZ+(W(K)*Z(K))
      SUMWW=SUMWW+(W(K)*W(K))
      SUMZZ=SUMZZ+(Z(K)*Z(K))
   30 CONTINUE
C
      N=N-M
C
      S=FLOAT(N)
      TAU=((SUMWZ*S)-(SUMW*SUMZ))/((SUMWW*S)-(SUMW*SUMW))
      ATAU=ABS(TAU)
      HIT=((SUMWW*SUMZ)-(SUMWZ*SUMW))/(((SUMWW*S)-(SUMW*SUMW)))
      DO 888 J=M,N
      RES=(Z(J)-(HIT+TAU*W(J)))**2
      TRES=RES+TRES
  888 CONTINUE
      U2=SUMWW-(SUMW*SUMW/N)
      V2=SUMZZ-(SUMZ*SUMZ/N)
      SYX=(V2-(TAU*TAU*U2))/(N-2)
      VSQ=SYX/U2
      CL=ABS(VSQ*TAS)
      HIT=HIT*XD/(XL*ATAU)
C
      WRITE(6,35)ATAU,CL
   35 FORMAT(1H0,'TAU= ',F10.3,'      95% CONFIDENCE LIMITS = +/-',F10.3,
     $/,1X)
C
      WRITE(6,36)HIT
   36 FORMAT(1H0,'COEFFICENT A IS ...    ',F10.5,/,1X)
      WRITE(6,37)
   37 FORMAT(1H0,'TAU IS EQUAL TO THE NEGATIVE SLOPE WHERE W(K) IS THE X
     * COORDINATE AND Z(K) IS THE Y COORDINATE.')
C
C
C
      WRITE(6,103)
  103 FORMAT(1H0,'TIME BETWEEN CHANNELS IS 0.097 NSEC')
      WRITE(6,104)TRES
  104 FORMAT(1H1,'PHASE-PLANE CORRELATION',28X,'Z(K) VS. W(K)',10X,'RESI
     $DUAL= ',E10.3)
      LL=N/100
      IF(LL.LT.1)LL=1
```

```
00000000011111111112222222222333333333344444444445555555555666666666677777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890
      DO 612 LA=M,N,LL
      I=I+1
      IF(I.GT.100)GO TO 88
      X(I)=W(LA)
      Y(I)=Z(LA)
  612 CONTINUE
      GO TO 89
   88 I=I-1
   89 CALL FRAMP(X,Y,I,6)
      CALL RPLOT(F,D,TSPAN,N,ATAU,HIT)
C
      STOP
      END
      SUBROUTINE FRAMP (X,Y,NPTS,KW)
C
C COMPUTES XMAX, ETC., AND CALLS FPLOT.
C
      DIMENSION X(101),Y(101)
      NCOLS=101
      NROWS=51
      XMAX=X(1)
      XMIN=X(1)
      YMAX=Y(1)
      YMIN=Y(1)
      DO 7070 J=1,NPTS
        IF(X(J)-XMAX)7010,7010,7000
 7000   XMAX=X(J)
 7010   IF(X(J)-XMIN)7020,7030,7030
 7020   XMIN=X(J)
 7030   IF(Y(J)-YMAX)7050,7050,7040
 7040   YMAX=Y(J)
 7050   IF(Y(J)-YMIN)7060,7070,7070
 7060   YMIN=Y(J)
 7070   CONTINUE
      CALL FPLOT (X,Y,NPTS,XMIN,XMAX,YMIN,YMAX,KW,NCOLS,NROWS)
      RETURN
      END
      SUBROUTINE FPLOT (X,Y,NPTS,XLEFT,XRITE,YBOTM,YTOP,KW,NCOLS,NROWS)
C
C
C PORTABLE SUBROUTINE FOR PRODUCING SIMPLE GRAPHS ON A LINE PRINTER.
C
C     X(J)     -   ABSCISSA OF THE J-TH POINT
C     Y(J)     --  ORDINATE OF THE J-TH POINT
C     NPTS     --  NUMBER OF POINTS
C     XLEFT, ETC.  --  BOUNDARIES OF THE GRAPH
C     KW       --  LOGICAL UNIT NUMBER OF THE LINE PRINTER
C     NCOLS    --  NUMBER OF COLUMNS (PRINT POSITIONS) IN THE PLOT
C                        (NCOLS MUST BE .LE. 101)
C     NROWS    --  NUMBER OF LINES IN THE PLOT
C
C THE ORDER OF THE POINTS IS SCRAMBLED BY FPLOT.  HOWEVER, THE
C CORRESPONDENCE BETWEEN X(J) AND Y(J) IS PRESERVED.
C
      DIMENSION X(101),Y(101)
```

```
000000000111111111122222222223333333333444444444455555555556666666666777777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890
CARD
      DIMENSION KLINE(103),KH(10),XX(6)
      DIMENSION DXDY(2),NCRPL(2),XLYB(2),XRYT(2),XYSGN(2),JXYAX(2)
      EQUIVALENCE (DX,DXDY(1)),(DY,DXDY(2)),(NCPLU,NCRPL(1)),
     *      (NRPLU,NCRPL(2)),(XL,XLYB(1)),(YB,XLYB(2)),(XR,XRYT(1)),
     *      (YT,XRYT(2)),(XSIGN,XYSGN(1)),(YSIGN,XYSGN(2)),
     *      (JCYAX,JXYAX(1)),(JRXAX,JXYAX(2))
      DATA KPL/1H+/, KMI/1H-/, KI/1HI/, KBL/1H /
      DATA KH(1),KH(2),KH(3),KH(4),KH(5),KH(6),KH(7),KH(8),KH(9),KH(10)
     *      /1H*,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9,1HU/
C
C NXD AND NYD ARE THE NUMBERS OF SPACES BETWEEN PRINTED VALUES OF X AND
C Y, RESPECTIVELY.  IF NXD.NE.20, THE LAST FORMAT STATEMENT IN
C THIS SUBROUTINE MUST BE ALTERED ACCORDINGLY.
      NXD=20
      NYD=10
C
C KPMAX IS THE DIMENSION OF KH(*).
      KPMAX=10
      NTRY=100
      UNITR=1
      RTWO=2
      RHALF=UNITR/RTWO
      RTWOH=RTWO+RHALF
      XL=XLEFT
      XR=XRITE
      YB=YBOTM
      YT=YTOP
      NCPLU=NCOLS+1
      NCPLT=NCPLU+1
      NRPLU=NROWS+1
      NRPLT=NRPLU+1
C
C COMPUTE DX AND DY, AND CHECK THE VALUES OF XL, ETC.
      DO 4080 J=1,2
        ENCRM=NCRPL(J)-2
        DO 4070 K=1,NTRY
          DXDY(J)=(XRYT(J)-XLYB(J))/ENCRM
          IF(DXDY(J))4080,4000,4080
 4000     IF(K-(NTRY-1))4010,4060,4630
 4010     IF(XLYB(J))4030,4020,4030
 4020     IF(XRYT(J))4030,4060,4030
 4030     IF(ABS(XLYB(J))-ABS(XRYT(J)))4050,4050,4040
 4040     XLYB(J)=XLYB(J)*RTWO
          GO TO 4070
 4050     XRYT(J)=XRYT(J)*RTWO
          GO TO 4070
 4060     XLYB(J)=-UNITR
          XRYT(J)=UNITR
 4070     CONTINUE
 4080   CONTINUE
C
C COMPUTE JCYAX, JRXAX, XSIGN, AND YSIGN.
      JCYAX=-XL/DX+RTWOH
      JRXAX=YT/DY+RTWOH
      DO 4150 J=1,2
```

```
0000000001111111111222222222233333333334444444444555555555566666666667777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890

           XYSGN(J)=UNITR
           IF(DXDY(J))4090,4100,4100
   4090    XYSGN(J)=-UNITR
   4100    IF(XLYB(J)*XYSGN(J))4110,4110,4140
   4110    IF(XRYT(J)*XYSGN(J))4140,4120,4120
   4120    IF(JXYAX(J)-2)4140,4130,4130
   4130    IF(JXYAX(J)-NCRPL(J))4150,4150,4140
   4140    JXYAX(J)=0
   4150    CONTINUE
C
C SORT THE POINTS ON Y, FROM TOP TO BOTTOM.
        CALL PSORT (X,Y,1,NPTS,1,-YSIGN)
C
C IGNORE ANY POINTS THAT ARE ABOVE THE TOP BOUNDARY.
        JP=1
        DO 4160 J=1,NPTS
           IF((Y(J)-YT)*YSIGN)4170,4170,4160
   4160    JP=JP+1
C
C LOOP OVER THE NRPLT LINES OF THE PLOT.
   4170 DO 4600 JR=1,NRPLT
C
C SET UP THE JR-TH PRINT LINE.
           JRMT=JR-2
           KFRST=KI
           KDIV=KBL
           KINBE=KBL
           JTEST=JRMT-(JRMT/NYD)*NYD
           IF(JRMT)4190,4180,4180
   4180    IF(JR-NRPLT)4200,4190,4190
   4190    JTEST=-1
           KFRST=KBL
           KDIV=KPL
           KINBE=KMI
           GO TO 4220
   4200    IF(JTEST)4220,4210,4220
   4210    KFRST=KPL
   4220    DO 4230 JC=2,NCPLU
   4230       KLINE(JC)=KINBE
           KLINE(1)=KFRST
           KLINE(NCPLU+1)=KFRST
           IF(JTEST)4240,4260,4260
   4240    DO 4250 JC=2,NCPLU,NXD
   4250       KLINE(JC)=KDIV
           GO TO 4580
C
C FIND THE POINTS, JFRST THROUGH JLAST, WHICH BELONG ON THIS LINE.
C THE TEST USING JY HAS BETTER ROUNDOFF PROPERTIES THAN DO
C SOME ALTERNATIVE TESTS.
   4260    JFRST=JP
   4270    IF(JP-NPTS)4280,4280,4310
   4280    IF((Y(JP)-YB)*YSIGN)4310,4290,4290
   4290    JY=(YT-Y(JP))/DY+RTWOH
           IF(JY-JR)4300,4300,4310
   4300    JP=JP+1
```

```
00000000011111111112222222222333333333344444444445555555555666666666677777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890
CARD
           GO TO 4270
   4310    JLAST=JP-1
C
C PUT IN THE X AND Y AXES.
           KDIV=KI
           IF(JR-JRXAX)4340,4320,4340
   4320    DO 4330 JC=2,NCPLU
   4330      KLINE(JC)=KMI
           KDIV=KPL
   4340    IF(JCYAX)4360,4360,4350
   4350    KLINE(JCYAX)=KDIV
   4360    IF(JLAST-JFRST)4530,4380,4370
C
C SORT THE POINTS ON THIS LINE ON X, FROM LEFT TO RIGHT.
   4370    CALL PSORT (X,Y,JFRST,JLAST,0,XSIGN)
C
C FIND THE NUMBER OF POINTS WHICH FALL INTO EACH PRINT POSITION
   4380    JX=JFRST
   4390    KP=0
           JC=0
   4400    JCSAV=JC
           IF((X(JX)-XL)*XSIGN)4460,4410,4410
   4410    IF((X(JX)-XR)*XSIGN)4420,4420,4460
   4420    JC=(X(JX)-XL)/DX+RTWOH
           IF(JC-JCSAV)4450,4450,4430
   4430    IF(KP)4470,4440,4470
   4440    JCSAV=JC
   4450    KP=KP+1
   4460    JX=JX+1
           IF(JX-JLAST)4400,4400,4470
C
C PLACE THE CORRESPONDING CHARACTER INTO THE APPROPRIATE PRINT POSITION.
   4470    IF(KP-KPMAX)4490,4490,4480
   4480    KP=KPMAX
   4490    IF(JCSAV-2)4520,4500,4500
   4500    IF(JCSAV-NCPLU)4510,4510,4520
   4510    KLINE(JCSAV)=KH(KP)
   4520    IF(JX-JLAST)4390,4390,4530
C
C THE PLOT LINE IS NOW SET.  COMPUTE A Y LABEL IF NECESSARY, AND PRINT.
   4530    IF(JTEST)4580,4540,4580
   4540    YY=JRMT
           YY=YT-YY*DY
           IF(JR-NROWS)4560,4560,4550
   4550    YY=YB
   4560    WRITE(KW,4570)YY,(KLINE(J),J=1,NCPLT)
   4570    FORMAT(1X,E14.7,1X,103A1)
           GO TO 4600
   4580    WRITE(KW,4590)(KLINE(J),J=1,NCPLT)
   4590    FORMAT(16X,103A1)
   4600    CONTINUE
C
C PRINT X LABELS.
        NXX=(NCOLS-1)/NXD+1
        DX=NXX-1
```

```
00000000011111111112222222222333333333344444444445555555555666666666677777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890
CARD
      DX=(XR-XL)/DX
      DO 4610 JX=1,NXX
        AJ=JX-1
 4610   XX(JX)=XL+AJ*DX
      XX(NXX)=XR
      WRITE(KW,4620)(XX(JX),JX=1,NXX)
 4620 FORMAT(/6(1X,E19.7))
 4630 RETURN
      END
      SUBROUTINE PSORT (X,Y,JFRST,JLAST,MODE,XYSGN)
C
C SORTS POINTS FOR SUBROUTINE FPLOT, USING SHELL-S METHOD....
C
C SORTS POINTS NUMBER -JFRST- THROUGH -JLAST-, INCLUSIVE.
C IF MODE.EQ.0, THE POINTS ARE SORTED ON X, OTHERWISE ON Y.
C IF XYSGN.GT.0, THE POINTS ARE SORTED INTO ASCENDING ORDER,
C OTHERWISE INTO DESCENDING ORDER.
C
      DIMENSION X(101),Y(101)
C
      N=JLAST-JFRST+1
      IF(N-1)5090,5090,5000
 5000 KBASE=JFRST-1
C
C COMPUTE THE INITIAL INCREMENT, M.
      L=2
 5010 L=L+L
      IF(L-N)5010,5010,5020
 5020 M=(L-1)/2
C
C DO M INSERTION SORTS.
 5030 K=N-M
      DO 5080 J=1,K
        L=KBASE+J
 5040   LPM=L+M
C
C COMPARE POINTS NUMBER LPM AND L.
        IF(MODE)5060,5050,5060
 5050   IF((X(LPM)-X(L))*XYSGN)5070,5080,5080
 5060   IF((Y(LPM)-Y(L))*XYSGN)5070,5080,5080
C
C THE POINTS ARE OUT OF ORDER.  SWAP THEM.
 5070   TEMP=X(L)
        X(L)=X(LPM)
        X(LPM)=TEMP
        TEMP=Y(L)
        Y(L)=Y(LPM)
        Y(LPM)=TEMP
        L=L-M
        IF(L-JFRST)5080,5040,5040
 5080   CONTINUE
C
C DIMINISH THE INCREMENT, M
      M=M/2
      IF(M)5090,5090,5030
```

```
         00000000011111111112222222222333333333344444444445555555555666666666677777777778
         12345678901234567890123456789012345678901234567890123456789012345678901234567890
5090 RETURN
      END
      SUBROUTINE RPLOT(F,D,FACT,NCHAN,ATAU,HIT)
      DIMENSION XARRAY(1026)
      DIMENSION WORKSP(6000)
      DIMENSION Y(1026),P(1026)
      DIMENSION IA(4)
      DIMENSION D(1024),F(1024)
C
      INTEGER LIFE(2),IT1(2)
      INTEGER IBCB(3),IBCD(3)
C
C
      DATA LIFE(1),LIFE(2)/'LIFE','TIME'/
      DATA IA1,IT1(1),IT1(2)/'A1 =','TAU1',' ='/
      DATA IA(1),IA(2),IA(3),IA(4)/' PHA','SE-P','LANE',' '/
      DATA IBCD(1),IBCD(2),IBCD(3)/'TIME',' NSE','C'/
      DATA IBCB(1),IBCB(2),IBCB(3)/'INTE','NSIT','Y'/
C
      TAU1=ATAU
      A1=HIT
C
      DO 909 I=1,NCHAN
      Y(I)=D(I)
      P(I)=F(I)
  909 CONTINUE
C
      XARRAY(1)=0.0
      DO 20 I=2,NCHAN
      XARRAY(I)=XARRAY(I-1)+FACT
   20 CONTINUE
C
      CALL PLOTS(WORKSP,6000,15)
      CALL PLOT(0.0,0.0,-3)
      CALL PLOT(0.0,8.0,2)
      CALL PLOT(11.,8.0,2)
      CALL PLOT(11.,0.0,2)
      CALL PLOT(0.0,0.0,2)
      CALL SCALE(XARRAY,11.,NCHAN,1)
      CALL SCALE(P,7.0,NCHAN,1)
      CALL SCALE(Y,7.0,NCHAN,1)
      CALL LINE(XARRAY,P,NCHAN,1,-1,83)
      CALL PLOT(0.0,0.0,3)
      CALL PLOT(0.0,0.0,2)
      CALL LINE(XARRAY,Y,NCHAN,1,-1,68)
      CALL AXIS(1.0,7.9,IBCD,-9,9.0,0.0,10.0,XARRAY(NCHAN+2))
      CALL SYMBOL(1.0,0.3,0.7,IBCB,90.0,9)
      CALL SYMBOL(1.5,6.8,0.7,IA,0.0,13)
      CALL SYMBOL(8.0,5.8,0.28,LIFE,0.0,8)
      CALL SYMBOL(8.0,5.4,0.28,IA1,0.0,4)
      CALL SYMBOL(8.0,4.8,0.28,IT1,0.0,6)
      CALL NUMBER(9.0,5.4,0.28,A1,0.0,5)
      CALL NUMBER(9.0,4.8,0.28,TAU1,0.0,5)
      CALL PLOT(10.0,0.0,999)
      RETURN
```

```
00000000011111111112222222222333333333344444444445555555555666666666677777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890
      END
      SUBROUTINE UERTST(IER,NAME)
      INTEGER IER
      INTEGER*2 NAME(3)
      INTEGER*2 NAMSET(3),NAMEQ(3)
      DATA NAMSET/2HUE,2HRS,2HET/
      DATA NAMEQ/2H  ,2H  ,2H  /
      DATA LEVEL/4/,IEQDF/0/,IEQ/1H=/
      IF(IER.GT.999)GO TO 25
      IF(IER.LT.-32)GO TO 55
      IF(IER.LE.128)GO TO 5
      IF(LEVEL.LT.1)GO TO 30
      CALL UGETIO(1,NIN,IOUNIT)
      IF(IEQDF.EQ.1)WRITE(IOUNIT,35)IER,NAMEQ,IEQ,NAME
      IF(IEQDF.EQ.0)WRITE(IOUNIT,35)IER,NAME
      GO TO 30
    5 IF(IER.LE.64)GO TO 10
      IF(LEVEL.LT.2)GO TO 30
      CALL UGETIO(1,NIN,IOUNIT)
      IF(IEQDF.EQ.1)WRITE(IOUNIT,40)IER,NAMEQ,IEQ,NAME
      IF(IEQDF.EQ.0)WRITE(IOUNIT,40)IER,NAME
      GO TO 30
   10 IF(IER.LE.32)GO TO 15
      IF(LEVEL.LT.3)GO TO 30
      CALL UGETIO(1,NIN,IOUNIT)
      IF(IEQDF.EQ.1)WRITE(IOUNIT,45)IER,NAMEQ,IEQ,NAME
      IF(IEQDF.EQ.0)WRITE(IOUNIT,45)IER,NAME
      GO TO 30
   15 CONTINUE
      DO 20 I=1,3
      IF(NAME(I).NE.NAMSET(I))GO TO 25
   20 CONTINUE
      LEVOLD=LEVEL
      LEVEL=IER
      IER-LEVOLD
      IF(LEVEL.LT.0)LEVEL=4
      IF(LEVEL.GT.4)LEVEL=4
      GO TO 30
   25 CONTINUE
      IF(LEVEL.LT.4)GO TO 30
      CALL UGETIO(1,NIN,IOUNIT)
      IF(IEQDF.EQ.1)WRITE(IOUNIT,50)IER,NAMEQ,IEQ,NAME
      IF(IEQDF.EQ.0)WRITE(IOUNIT,50)IER,NAME
   30 IEQDF=0
      RETURN
   35 FORMAT(19H *** TERMINAL ERROR,10X,7H(IER = ,I3,20H) FROM IMSL ROUT
     *INE ,3A2,A1,3A2)
   40 FORMAT(36H *** WARNING WITH FIX ERROR  (IER = ,I3,20H) FROM IMSL R
     *OUTINE ,3A2,A1,3A2)
   45 FORMAT(18H *** WARNING ERROR,11X,7H(IER = ,I3,20H) FROM IMSL ROUTI
     *NE ,3A2,A1,3A2)
   50 FORMAT(20H *** UNDEFINED ERROR,9X,7H(IER = ,I5,20H) FROM IMSL ROUT
     *INE ,3A2,A1,3A2)
   55 IEQDF=1
      DO 60 I=1,3
```

```
     00000000011111111112222222222333333333344444444445555555555666666666677777777778
     12345678901234567890123456789012345678901234567890123456789012345678901234567890
CARD
      60 NAMEO(I)=NAME(I)
      65 RETURN
         END
         SUBROUTINE UGETIO(IOPT,NIN,NOUT)
         INTEGER IOPT,NIN,NOUT
         INTEGER NIND,NOUTD
         DATA NIND/5/,NOUTD/6/
         IF(IOPT.EQ.3)GO TO 10
         IF(IOPT.EQ.2)GO TO 5
         IF(IOPT.NE.1)GO TO 9005
         NIN=NIND
         NOUT=NOUTD
         GO TO 9005
       5 NIND=NIN
         GO TO 9005
      10 NOUTD=NOUT
    9005 RETURN
         END
         SUBROUTINE DCSQDU(X,Y,NX,C,IC,A,B,Q,IER,FS)
         INTEGER NX,IC,IER
         REAL X(NX),Y(NX),C(IC,3),A,B,Q
         REAL FS(IC)
         INTEGER I,IA,IB,IBM1,IPT,IV,JER,KER,NXM1
         REAL D,DA,DB,DD,DX,FOURTH,HALF,QA,QAB,QB
         REAL SIXTH,THIRD,V,ZERO
         DATA SIXTH/0.1666667/,THIRD/0.3333333/
         DATA ZERO/0.0/,FOURTH/0.25/,HALF/0.5/
         JER=0
         KER=0
         NXM1=NX-1
         IPT=1
         IA=1
         V=AMIN1(A,B)
       5 D=V-X(IA)
         DO 10 I=IA,NXM1
         IV=I
         DD=V-X(I+1)
         IF(DD.LT.ZERO)GO TO 15
         IF(I.LT.NXM1)D=DD
      10 CONTINUE
         IV=NXM1
         IF(DD.GT.ZERO)KER=34
      15 CONTINUE
         IF(D.LT.ZERO)JER=33
         IF(IPT.EQ.2)GO TO 20
         IPT=2
         IA=IV
         DA=D
         V=AMAX1(A,B)
         GO TO 5
      20 IB=IV
         DB=D
         QA=(((FOURTH*C(IA,3)*DA+THIRD*C(IA,2))*DA+HALF*C(IA,1))*DA+Y(IA))*
        $DA
         QAB=ZERO
```

```
00000000011111111112222222222333333333344444444445555555555666666666677777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890
      IBM1=IB-1
      IF(IBM1.LT.IA)GO TO 30
      DO 25 I=IA,IBM1
      DX=X(I+1)-X(I)
      QAB=QAB+HALF*DX*(Y(I+1)+Y(I)-(C(I+1,2)+C(I,2))*DX*DX*SIXTH)
      FS(I)=QAB
   25 CONTINUE
   30 QB=(((FOURTH*C(IB,3)*DB+THIRD*C(IB,2))*DB+HALF*C(IB,1))*DB+Y(IB))*
     $DB
      Q=QB+QAB-QA
      IF(B.LT.A)Q=-Q
      IER=MAX0(JER,KER)
 9000 CONTINUE
      IF(JER.GT.0)CALL UERTST(JER,6HDCSQDU)
      IF(KER.GT.0)CALL UERTST(KER,6HDCSQDU)
 9005 RETURN
      END
      SUBROUTINE ICSICU(X,Y,NX,BPAR,C,IC,IER)
      INTEGER NX,IC,IER
      REAL X(NX),Y(NX),BPAR(4),C(IC,3)
      INTEGER I,J,NXM1
      REAL DX,DXJ,DXJP1,DXP,DYJ,DYJP1,HALF,ONE,PJ
      REAL SIX,SIXI,TWO,YPPA,YPPB,ZERO
      EQUIVALENCE (DXJ,YPPB),(PJ,SIXI),(DXJP1,YPPA)
      DATA ZERO/0.0/,HALF/0.5/,ONE/1.0/
      DATA TWO/2.0/,SIX/6.0/
      IER=0
      NXM1=NX-1
      IF(IC.LT.NXM1)GO TO 30
      IF(NX.LT.2)GO TO 35
      IF(NX.EQ.2)GO TO 10
      DXJ=X(2)-X(1)
      IF(DXJ.LE.ZERO)GO TO 40
      DYJ=Y(2)-Y(1)
      DO 5 J=2,NXM1
      DXJP1=X(J+1)-X(J)
      IF(DXJP1.LE.ZERO) GO TO 40
      DYJP1=Y(J+1)-Y(J)
      DXP=DXJ+DXJP1
      C(J,1)=DXJP1/DXP
      C(J,2)=ONE-C(J,1)
      C(J,3)=SIX*(DYJP1/DXJP1-DYJ/DXJ)/DXP
      DXJ=DXJP1
      DYJ=DYJP1
    5 CONTINUE
   10 C(1,1)=-BPAR(1)*HALF
      C(1,2)=BPAR(2)*HALF
      IF(NX.EQ.2)GO TO 20
      DO 15 J=2,NXM1
      PJ=C(J,2)*C(J-1,1)+TWO
      C(J,1)=-C(J,1)/PJ
      C(J,2)=(C(J,3)-C(J,2)*C(J-1,2))/PJ
   15 CONTINUE
   20 YPPB=(BPAR(4)-BPAR(3)*C(NXM1,2))/(BPAR(3)*C(NXM1,1)+TWO)
      SIXI=ONE/SIX
```

```
00000000011111111112222222222333333333344444444445555555555666666666677777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890
      DO 25 I=1,NXM1
      J=NX-I
      YPPA=C(J,1)*YPPB+C(J,2)
      IF(YPPA.LT.1E-25)YPPA=0.0
      DX=X(J+1)-X(J)
      C(J,3)=SIXI*(YPPB-YPPA)/DX
      C(J,2)=HALF*YPPA
      IF(C(J,2).LE.1E-25)C(J,2)=0.0
      IF(C(J,3).LT.1E-25)C(J,3)=0.0
      R=Y(J+1)-Y(J)
      IF(R.LT.1E-25)R=0.0
      C(J,1)=R/DX-(C(J,2)+C(J,3)*DX)*DX
      YPPB=YPPA
   25 CONTINUE
      GO TO 9005
   30 IER=129
      GO TO 9000
   35 IER=130
      GO TO 9000
   40 IER=131
 9000 CONTINUE
      CALL UERTST(IER,6HICSICU)
 9005 RETURN
      END
//LKED.SYSLMOD DD DSNAME=OSU.ACT13530.PROGRAM(PHASE1),DISP=OLD
//LKED.LIB DD DSN=SYS7.COMPLOT.PRTPLT,DISP=SHR
//LKED.SYSIN DD *
  INCLUDE LIB(PLOTS)
//
$ENDLIST
```

```
00000000011111111112222222222333333333344444444445555555555666666666677777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890
//OKLAHOMA JOB (13530,441-62-4034),'BOB',TIME=(0,40),CLASS=A
//  EXEC FORTGCL
//FORT.SYSIN DD *
C PHASE-PLANE METHOD
C  USING LEAST SQUARES
C
C
C
C WHERE M IS THE INITIAL POINT FOR LEAST SQUARES
C AND TSPAN IS THE INSTANTANEOUS TIME DIFFERENTIAL
C AND
C
C Z(K)=DSUM/FSUM=FS(K)/DS(K)
C
C W(K)=D(K)/DSUM=D(K)/DS(K)
C
C AND WHERE,
C
C Z(T)=(-TAU*W(K)+K*TAU)
C
C K IS EQUAL TO HIT
C
C
      DIMENSION R(1024),C(1024,3)
      DIMENSION FS(1024),DS(1024)
      DIMENSION Z(1024),W(1024)
      DIMENSION D(1024),F(1024)
      DIMENSION BPAR(4)
C
C INITIALIZE VARIABLES
      TRES=0.0
      I=0
      SS=0.0
      SUMWW=0.
      SUMWZ=0.
      SUMZ=0.
      SUMW=0.
      SUMZZ=0.0
      XL=0.0
      XD=0.0
      TAS=1.96
      IC=1024
      BPAR(1)=0.0
      BPAR(2)=0.0
      BPAR(3)=0.0
      BPAR(4)=0.0
C
C INITIALIZE OUTPUT ARRAY
C
C
C
C
      TSPAN=0.097
```

```
00000000011111111112222222222333333333344444444445555555555666666666677777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890
      R(1)=TSPAN
      A=R(1)
C
C READ VARIABLES
      WRITE(6,762)
  762 FORMAT(1HO,'READ IN NUMBER OF CHANNELS (I4)')
      READ(5,300)N
  300 FORMAT(I4)
      NN=N
      NNCHAN=N/10
      DO 752 J=1,NNCHAN
      L=J*10-9
      LLL=L+9
      READ(1,11)(F(K),K=L,LLL)
  752 CONTINUE
      DO 753 J=1,NNCHAN
      L=J*10-9
      LLL=L+9
      READ(2,11)(D(K),K=L,LLL)
      DO 616 II=L,LLL
      IF(SS.LT.D(II))M=II
      IF(SS.LT.D(II))SS=D(II)
  616 CONTINUE
  753 CONTINUE
   11 FORMAT(10(1X,F6.0))
      WRITE(6,45)
   45 FORMAT(1H1,'PHASE-PLANE METHOD USING LEAST SQUARES ')
C
C
C  NORMALIZE DECAY CURVES
C
      DO 660 J=1,NN
      IF(F(J).GT.XL)XL=F(J)
      IF(D(J).GT.XD)XD=D(J)
  660 CONTINUE
      DO 665 J=1,NN
      D(J)=D(J)*XL/XD
  665 CONTINUE
      DO 22 K=2,NN
      R(K)=R(K-1)+TSPAN
   22 CONTINUE
      B=R(NN)
C
C
C CALCULATE SUMS
      CALL ICSICU(R,F,NN,BPAR,C,IC,IER)
      CALL DCSQDU(R,F,NN,C,IC,A,B,Q,IER,FS)
      CALL ICSICU(R,D,NN,BPAR,C,IC,IER)
      CALL DCSQDU(R,D,NN,C,IC,A,B,Q,IER,DS)
      DO 21 K=1,NN
      IF(FS(K).LE.0.0)Z(K)=0.0
      IF(FS(K).LE.0.0)W(K)=0.0
      IF(FS(K).LE.0.0)GO TO 21
   15 Z(K)=DS(K)/FS(K)
      W(K)=D(K)/FS(K)
```

```
00000000011111111112222222222333333333344444444445555555555666666666677777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890
   21 CONTINUE
C
C
C  DO LEAST SQUARES
C
      NN=N-1
      DO 30 K=M,NN
      SUMW=SUMW+W(K)
      SUMZ=SUMZ+Z(K)
      SUMWZ=SUMWZ+(W(K)*Z(K))
      SUMWW=SUMWW+(W(K)*W(K))
      SUMZZ=SUMZZ+(Z(K)*Z(K))
   30 CONTINUE
C
      N=NN-M
C
      S=FLOAT(N)
      TAU=((SUMWZ*S)-(SUMW*SUMZ))/((SUMWW*S)-(SUMW*SUMW))
      ATAU=ABS(TAU)
      HIT=((SUMWW*SUMZ)-(SUMWZ*SUMW))/(((SUMWW*S)-(SUMW*SUMW)))
      DO 888 J=M,N
      RES=(Z(J)-(HIT+TAU*W(J)))**2
      TRES=RES+TRES
  888 CONTINUE
      U2=SUMWW-(SUMW*SUMW/N)
      V2=SUMZZ-(SUMZ*SUMZ/N)
      SYX=(V2-(TAU*TAU*U2))/(N-2)
      VSQ=SYX/U2
      CL=ABS(VSQ*TAS)
      HIT=HIT*XD/(XL*ATAU)
C
      WRITE(6,35)ATAU,CL
   35 FORMAT(1HO,'TAU= ',F10.3,'      95% CONFIDENCE LIMITS = +/-',F10.3,
     $/,1X)
C
      WRITE(6,36)HIT
   36 FORMAT(1HO,'COEFFICENT A IS ...   ',F10.5,/,1X)
      WRITE(6,104)TRES
  104 FORMAT(1HO,'PHASE-PLANE CORRELATION',2X,'Z(K) VS. W(K)',2X,'RESIDU
     $AL= ',E10.3)
      WRITE(6,37)
   37 FORMAT(1HO,'TAU IS EQUAL TO THE NEGATIVE SLOPE')
      WRITE(6,554)
  554 FORMAT(1X,'WHERE W(K) IS THE X COORDINATE AND Z(K) IS')
      WRITE(6,555)
  555 FORMAT(1X,'THE Y COORDINATE.')
C
C
C
      WRITE(6,103)
  103 FORMAT(1HO,'TIME BETWEEN CHANNELS IS 0.097 NSEC')
C
      STOP
      END
      SUBROUTINE UERTST(IER,NAME)
```

```
000000000111111111122222222223333333333444444444455555555556666666666777777777778
123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
      INTEGER IER
      INTEGER*2 NAME(3)
      INTEGER*2 NAMSET(3),NAMEQ(3)
      DATA NAMSET/2HUE,2HRS,2HET/
      DATA NAMEQ/2H  ,2H  ,2H  /
      DATA LEVEL/4/,IEQDF/0/,IEQ/1H=/
      IF(IER.GT.999)GO TO 25
      IF(IER.LT.-32)GO TO 55
      IF(IER.LE.128)GO TO 5
      IF(LEVEL.LT.1)GO TO 30
      CALL UGETIO(1,NIN,IOUNIT)
      IF(IEQDF.EQ.1)WRITE(IOUNIT,35)IER,NAMEQ,IEQ,NAME
      IF(IEQDF.EQ.0)WRITE(IOUNIT,35)IER,NAME
      GO TO 30
    5 IF(IER.LE.64)GO TO 10
      IF(LEVEL.LT.2)GO TO 30
      CALL UGETIO(1,NIN,IOUNIT)
      IF(IEQDF.EQ.1)WRITE(IOUNIT,40)IER,NAMEQ,IEQ,NAME
      IF(IEQDF.EQ.0)WRITE(IOUNIT,40)IER,NAME
      GO TO 30
   10 IF(IER.LE.32)GO TO 15
      IF(LEVEL.LT.3)GO TO 30
      CALL UGETIO(1,NIN,IOUNIT)
      IF(IEQDF.EQ.1)WRITE(IOUNIT,45)IER,NAMEQ,IEQ,NAME
      IF(IEQDF.EQ.0)WRITE(IOUNIT,45)IER,NAME
      GO TO 30
   15 CONTINUE
      DO 20 I=1,3
      IF(NAME(I).NE.NAMSET(I))GO TO 25
   20 CONTINUE
      LEVOLD=LEVEL
      LEVEL=IER
      IER=LEVOLD
      IF(LEVEL.LT.0)LEVEL=4
      IF(LEVEL.GT.4)LEVEL=4
      GO TO 30
   25 CONTINUE
      IF(LEVEL.LT.4)GO TO 30
      CALL UGETIO(1,NIN,IOUNIT)
      IF(IEQDF.EQ.1)WRITE(IOUNIT,50)IER,NAMEQ,IEQ,NAME
      IF(IEQDF.EQ.0)WRITE(IOUNIT,50)IER,NAME
   30 IEQDF=0
      RETURN
   35 FORMAT(19H *** TERMINAL ERROR,10X,7H(IER = ,I3,20H) FROM IMSL ROUT
     *INE ,3A2,A1,3A2)
   40 FORMAT(36H *** WARNING WITH FIX ERROR  (IER = ,I3,20H) FROM IMSL R
     *OUTINE ,3A2,A1,3A2)
   45 FORMAT(18H *** WARNING ERROR,11X,7H(IER = ,I3,20H) FROM IMSL ROUTI
     *NE ,3A2,A1,3A2)
   50 FORMAT(20H *** UNDEFINED ERROR,9X,7H(IER = ,I5,20H) FROM IMSL ROUT
     *INE ,3A2,A1,3A2)
   55 IEQDF=1
      DO 60 I=1,3
   60 NAMEQ(I)=NAME(I)
   65 RETURN
```

```
00000000011111111112222222222333333333344444444445555555555666666666677777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890
      END
      SUBROUTINE UGETIO(IOPT,NIN,NOUT)
      INTEGER IOPT,NIN,NOUT
      INTEGER NIND,NOUTD
      DATA NIND/5/,NOUTD/6/
      IF(IOPT.EQ.3)GO TO 10
      IF(IOPT.EQ.2)GO TO 5
      IF(IOPT.NE.1)GO TO 9005
      NIN=NIND
      NOUT=NOUTD
      GO TO 9005
    5 NIND=NIN
      GO TO 9005
   10 NOUTD=NOUT
 9005 RETURN
      END
      SUBROUTINE ICSICU(X,Y,NX,BPAR,C,IC,IER)
      INTEGER NX,IC,IER
      REAL X(NX),Y(NX),BPAR(4),C(IC,3)
      INTEGER I,J,NXM1
      REAL DX,DXJ,DXJP1,DXP,DYJ,DYJP1,HALF,ONE,PJ
      REAL SIX,SIXI,TWO,YPPA,YPPB,ZERO
      EQUIVALENCE (DXJ,YPPB),(PJ,SIXI),(DXJP1,YPPA)
      DATA ZERO/0.0/,HALF/0.5/,ONE/1.0/
      DATA TWO/2.0/,SIX/6.0/
      IER=0
      NXM1=NX-1
      IF(IC.LT.NXM1)GO TO 30
      IF(NX.LT.2)GO TO 35
      IF(NX.EQ.2)GO TO 10
      DXJ=X(2)-X(1)
      IF(DXJ.LE.ZERO)GO TO 40
      DYJ=Y(2)-Y(1)
      DO 5 J=2,NXM1
      DXJP1=X(J+1)-X(J)
      IF(DXJP1.LE.ZERO) GO TO 40
      DYJP1=Y(J+1)-Y(J)
      DXP=DXJ+DXJP1
      C(J,1)=DXJP1/DXP
      C(J,2)=ONE-C(J,1)
      C(J,3)=SIX*(DYJP1/DXJP1-DYJ/DXJ)/DXP
      DXJ=DXJP1
      DYJ=DYJP1
    5 CONTINUE
   10 C(1,1)=-BPAR(1)*HALF
      C(1,2)=BPAR(2)*HALF
      IF(NX.EQ.2)GO TO 20
      DO 15 J=2,NXM1
      PJ=C(J,2)*C(J-1,1)+TWO
      C(J,1)=-C(J,1)/PJ
      C(J,2)=(C(J,3)-C(J,2)*C(J-1,2))/PJ
   15 CONTINUE
   20 YPPB=(BPAR(4)-BPAR(3)*C(NXM1,2))/(BPAR(3)*C(NXM1,1)+TWO)
      SIXI=ONE/SIX
      DO 25 I=1,NXM1
```

```
0000000001111111111222222222233333333334444444444555555555566666666667777777778
1234567890123456789012345678901234567890123456789012345678901234567890123456789
CARD
      J=NX-1
      YPPA=C(J,1)*YPPB+C(J,2)
      IF(YPPA.LT.1E-25)YPPA=0.0
      DX=X(J+1)-X(J)
      C(J,3)=SIXI*(YPPB-YPPA)/DX
      C(J,2)=HALF*YPPA
      IF(C(J,2).LE.1E-25)C(J,2)=0.0
      IF(C(J,3).LT.1E-25)C(J,3)=0.0
      R=Y(J+1)-Y(J)
      IF(R.LT.1E-25)R=0.0
      C(J,1)=R/DX-(C(J,2)+C(J,3)*DX)*DX
      YPPB=YPPA
   25 CONTINUE
      GO TO 9005
   30 IER=129
      GO TO 9000
   35 IER=130
      GO TO 9000
   40 IER=131
 9000 CONTINUE
      CALL UERTST(IER,6HICSICU)
 9005 RETURN
      END
      SUBROUTINE DCSQDU(X,Y,NX,C,IC,A,B,Q,IER,FS)
      DIMENSION FS(1024)
      INTEGER NX,IC,IER
      REAL X(NX),Y(NX),C(IC,3),A,B,Q
      INTEGER I,IA,IB,IBM1,IPT,IV,JER,KER,NXM1
      REAL D,DA,DB,DD,DX,FOURTH,HALF,QA,QAB,QB
      REAL SIXTH,THIRD,V,ZERO
      DATA SIXTH/0.1666667/,THIRD/0.3333333/
      DATA ZERO/0.0/,FOURTH/0.25/,HALF/0.5/
      JER=0
      KER=0
      NXM1=NX-1
      IPT=1
      IA=1
      V=AMIN1(A,B)
    5 D=V-X(IA)
      DO 10 I=IA,NXM1
      IV=I
      DD=V-X(I+1)
      IF(DD.LT.ZERO)GO TO 15
      IF(I.LT.NXM1)D=DD
   10 CONTINUE
      IV=NXM1
      IF(DD.GT.ZERO)KER=34
   15 CONTINUE
      IF(D.LT.ZERO)JER=33
      IF(IPT.EQ.2)GO TO 20
      IPT=2
      IA=IV
      DA=D
      V=AMAX1(A,B)
      GO TO 5
```

```
000000000111111111122222222223333333333444444444455555555556666666666777777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890
     20 IB-IV
        DB=D
        OA=(((FOURTH*C(IA,3)*DA+THIRD*C(IA,2))*DA+HALF*C(IA,1))*DA+Y(IA))*
       $DA
        QAB=ZERO
        IBM1-IB-1
        IF(IBM1.LT.IA)GO TO 30
        DO 25 I=IA,IBM1
        DX=X(I+1)-X(I)
        QAB=QAB+HALF*DX*(Y(I+1)+Y(I)-(C(I+1,2)+C(I,2))*DX*DX*SIXTH)
        FS(I)=QAB
     25 CONTINUE
     30 QB=(((FOURTH*C(IB;3)*DB+THIRD*C(IB,2))*DB+HALF*C(IB,1))*DB+Y(IB))*
       $DB
        O=QB+QAB-QA
        IF(B.LT.A)O=-Q
        IER-MAXO(JER,KER)
   9000 CONTINUE
        IF(JER.GT.O)CALL UERISI(JER,6HDCSQDU)
        IF(KER.GT.O)CALL UERTST(KER,6HDCSQDU)
   9005 RETURN
        END
//LKED.SYSLMOD DD DSNAME=OSU.ACT13530.PROGRAM,DISP=OLD
//LKED.SYSIN DD *
  NAME PHASE2(R)
//
$ENDLIST
```

```
JOB LIST                                                     08:59:04.415 . 82/11/10  MVS/R3.7C

                                    80/80 LIST                               PAGE  001

        00000000011111111112222222222333333333344444444445555555555666666666677777777778
        12345678901234567890123456789012345678901234567890123456789012345678901234567890
CARD
0001    00000010ALLOCATE DATASET(LAMP.DATA) FILE(FT01F001) OLD
0002    00000020ALLOCATE DATASET(DECAY.DATA) FILE(FT02F001) OLD
0003    00000030ATTRIB ABC LRECL(80) BLKSIZE(6400) RECFM(F B)
0004    00000040ALLOCATE DATASET(RES.DATA) FILE(FT03F001) NEW USING(ABC)
0005    00000050ALLOCATE DATASET(CDATA.DATA) FILE(FT04F001) NEW USING(ABC)
0006    00000060CALL 'OSU.ACT13530.PROGRAM(OKLA3)'
0007    00000070FREE ATTR(ABC)
0008    $ENDLIST
```

```
$JOB LIST                                                  12:24:19.253  82/10/23  MVS/R3.7C
                                   80/80 LIST                              PAGE  001

         00000000011111111112222222222333333333344444444445555555555666666666677777777778
         12345678901234567890123456789012345678901234567890123456789012345678901234567890
CARD
0001     //OKLAHOMA JOB (12289,441-62-4034),'BOB',TIME=(0,40),CLASS=A
0002     //  EXEC PGM=PHASE1
0003     //STEPLIB DD DSN=OSU.ACT12288.PROGRAM,DISP=SHR
0004     //FT01F001 DD DSNAME=U12288A.LAMP5.DATA,DISP=SHR
0005     //FT02F001 DD DSNAME=U12288A.DECAY5.DATA,DISP=SHR
0006     //FT05F001 DD *
0007     1000
0008     //FT06F001 DD SYSOUT=A
0009     //FT15F001 DD SYSOUT=A,DCB=(RECFM=UA,BLKSIZE=133)
0010     //
0011     $ENDLIST
```

$JOB LIST

08:36:23.595 82/11/10 MVS/R3.7C

80/80 LIST

PAGE 001

```
     00000000011111111112222222222333333333344444444445555555555666666666677777777778
     12345678901234567890123456789012345678901234567890123456789012345678901234567890
CARD
0001 00000010PROC 4 LAMP DECAY RES CDATA
0002 00000020CONTROL NOMSG
0003 00000030FREE FILE(FT01F001 FT02F001 FT03F001 FT04F001)
0004 00000040FREE ATTR(ABC)
0005 00000050CONTROL MSG
0006 00000060ALLOCATE DATASET(&LAMP..DATA) FILE(FT01F001) OLD
0007 00000070ALLOCATE DATASET(&DECAY..DATA) FILE(FT02F001) OLD
0008 00000080ATTRIB ABC LRECL(80) BLKSIZE(6400) RECFM(F B)
0009 00000090ALLOCATE DATASET(&RES..DATA) FILE(FT03F001) NEW USING(ABC)
0010 00000100ALLOCATE DATASET(&CDATA..DATA) FILE(FT04F001) NEW USING(ABC)
0011 00000110CALL 'OSU.ACT13530.PROGRAM(OKLA3)'
0012 00000120FREE ATTR(ABC)
0013 00000130FREE FILE(FT01F001)
0014 00000140FREE FILE(FT02F001)
0015 00000150FREE FILE(FT03F001)
0016 00000160FREE FILE(FT04F001)
0017 $ENDLIST
```

12:24:19.253 82/10/23 MVS/R3.7C

$JOB LIST

80/80 LIST

PAGE 001

```
     00000000011111111112222222222333333333344444444445555555555666666666677777777778
     12345678901234567890123456789012345678901234567890123456789012345678901234567890
CARD
0001 //OKLAHOMA JOB (12289,441-62-4034),'BOB',TIME=(0,40),CLASS=A
0002 //  EXEC PGM=PHASE1
0003 //STEPLIB DD DSN=OSU.ACT12288.PROGRAM,DISP=SHR
0004 //FT01F001 DD DSNAME=U12288A.LAMP5.DATA,DISP=SHR
0005 //FT02F001 DD DSNAME=U12288A.DECAY5.DATA,DISP=SHR
0006 //FT05F001 DD *
0007 1000
0008 //FT06F001 DD SYSOUT=A
0009 //FT15F001 DD SYSOUT=A,DCB=(RECFM=UA,BLKSIZE=133)
0010 //
0011 $ENDLIST
```

Double

This program, primarily for manual operation, will execute OSU dataset program OKLA 2. OKLA 2 will not calculate or return a residual or autocorrelation file. When executed the command program will ask for lamp and decay dataset names. These files will be allocated and OKLA 2 called. All lifetime values from OKLA 2 will be displayed on the terminal.

DPRINT

This list program will execute OKLA 1 and print out a copy of the results in Math Sciences, Room 115. This program is primarily designed for manual operation.

The command list program works by building a batch job and submitting it. The output will appear under the name specified by the user. An account number and password must be known to use this program. Since this job submits a CNTL program, it is recommended that datasets not be named JOBSTREM. All datasets name JOBSTREM run the risk of being deleted.

DPRINT accesses three datasets. That is, a lamp dataset, a decay dataset, and a dataset with program parameters is accessed. These datasets can have almost any name. The program will ask for one to input these names as PROC parameters:LAMP, DECAY, and PDATA, respectively.

An example of the program execution and a PDATA input file is given on the following pages.

```
$JOB LIST                                              08:58:43.069 · 82/11/10  MVS/R3.7C

                                80/80 LIST                               PAGE  001

      00000000011111111112222222222333333333344444444445555555555666666666677777777778
      12345678901234567890123456789012345678901234567890123456789012345678901234567890
CARD
0001  00000010PROC 2 LAMP DECAY
0002  00000020ALLOCATE DATASET(&LAMP..DATA) FILE(FT01F001) OLD
0003  00000030ALLOCATE DATASET(&DECAY..DATA) FILE(FT02F001) OLD
0004  00000040CALL 'OSU.ACT12288.PROGRAM(OKLA2)'
0005  00000050FREE FILE(FT01F001)
0006  00000060FREE FILE(FT02F001)
0007  $ENDLIST
```

```
$JOB LIST                                                     08:37:39.141  82/11/10  MVS/R3.7C

                                         80/80 LIST                       PAGE  001

     00000000011111111112222222222333333333344444444445555555555666666666677777777778
     12345678901234567890123456789012345678901234567890123456789012345678901234567890
     00000010PROC 3 LAMP DECAY PDATA
     00000020/**********
     00000030/*  DOUBLE EXP PRINT
     00000040/**********
     00000050WRITE BATCH JOB DOUBLE EXP
     00000060WRITE FOR OSU.ACT13530.PROGRAM(OKLA1)
     00000070WRITE
     00000080WRITE PROCESSING ON &SYSDATE AT &SYSTIME
     00000090WRITE
     00000100WRITE ENTER THE JOBNAME
     00000110READ &JOBNAME
     00000120WRITE ENTER THE PROJECT NUMBER
     00000130READ &PROJ
     00000140WRITE ENTER THE PASSWORD
     00000150READ &PWORD
     00000160WRITE
     00000170WRITE JOB BUILT AND SUBMITTED
     00000180CONTROL NOMSG
     00000190DELETE JOBSTREM.CNTL
     00000200CONTROL MSG
     00000210ED JOBSTREM.CNTL NEW EMODE
     00000220010 //&JOBNAME JOB (&PROJ,441-62-4034),BOBK,TIME=(,40),CLASS=A,
     00000230020 //  MSGCLASS=A,NOTIFY=&SYSUID
     00000240030 ##PASSWORD &PWORD
     00000250040 ##ROUTE  PRINT LOCAL
     00000260050 //  EXEC PGM=OKLA1
     00000270060 //STEPLIB DD DSN=OSU.ACT13530.PROGRAM,DISP=SHR
     00000280070 //FT01F001 DD DSN=&SYSUID..&LAMP..DATA,DISP=SHR
     00000290080 //FT02F001 DD DSN=&SYSUID..&DECAY..DATA,DISP=SHR
     00000300090 //FT05F001 DD DSN=&SYSUID..&PDATA..DATA,DISP=SHR
     00000310100 //FT06F001 DD SYSOUT=A
     00000315105 //FT15F001 DD SYSOUT=A,DCB=(RECFM=UA,BLKSIZE=133)
     00000320110 //
     00000330C 10 99999 '##' '/*' ALL
     00000340SAVE
     00000350SUBMIT JOBSTREM.CNTL
     00000360END NOSAVE
     00000370CONTROL NOMSG
     00000380DELETE JOBSTREM.CNTL
     00000390CONTROL MSG
     $ENDLIST
```

Program Parameters - Example PDATA Datasheet

```
00010 BYN     - TITLE (17A4)
00020 1000    - NO OF CHANNELS (F4)
00030  0.0    - LAMP BACKGROUND (F5.2)
00040  0.0    - DECAY BACKGROUND (F5.2)
00050  400    - FITTING CHANNELS FROM (F4)
00060 1000    - TO (F4)
00070  1.0    - A1 GUESS
00080  1.0    - A2 GUESS
00090  4.0    - TAU1 GUESS
00100 10.0    - TAU2 GUESS
```

Phase P

This program will execute the phase-plane program for the deconvolution of a single exponential decay.

$JOB LIST 08:37:06.950 82/11/10 MVS/R3.7C

80/80 LIST PAGE 001

```
000000000111111111122222222223333333333444444444455555555556666666666777777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890
00000010PROC 3 LAMP DECAY NOPTS
00000020/**********
00000030/* PHASE PLANE PRINT
00000040/**********
00000050WRITE BATCH JOB SINGLE EXP
00000060WRITE FOR OSU.ACT13530.PROGRAM(PHASE1)
00000070WRITE
00000080WRITE PROCESSING ON &SYSDATE AT &SYSTIME
00000090WRITE
00000100WRITE ENTER THE JOBNAME
00000110READ &JOBNAME
00000120WRITE ENTER THE PROJECT NUMBER
00000130READ &PROJ
00000140WRITE ENTER THE PASSWORD
00000150READ &PWORD
00000160WRITE
00000170WRITE JOB BUILT AND SUBMITTED
00000180CONTROL NOMSG
00000190DELETE JOBSTREM.CNTL
00000200CONTROL MSG
00000210ED JOBSTREM.CNTL NEW EMODE
00000220010 //&JOBNAME JOB (&PROJ,441-62-4034),BOBK,TIME=(,40),CLASS=A,
00000230020 //   MSGCLASS=A,NOTIFY=&SYSUID
00000240030 ##PASSWORD &PWORD
00000250040 ##ROUTE  PRINT LOCAL
00000260050 //   EXEC PGM=PHASE1
00000270060 //STEPLIB DD DSN=OSU.ACT13530.PROGRAM,DISP=SHR
00000280070 //FT01F001 DD DSN=&SYSUID..&LAMP..DATA,DISP=SHR
00000290080 //FT02F001 DD DSN=&SYSUID..&DECAY..DATA,DISP=SHR
00000300090 //FT05F001 DD *
00000303093 &NOPTS
00000310100 //FT06F001 DD SYSOUT=A
00000315105 //FT15F001 DD SYSOUT=A,DCB=(RECFM=UA,BLKSIZE=133)
00000320110 //
00000330C 10 99999 '##' '/*' ALL
00000340SAVE
00000350SUBMIT JOBSTREM.CNTL
00000360END NOSAVE
00000370CONTROL NOMSG
00000380DELETE JOBSTREM.CNTL
00000390CONTROL MSG
$ENDLIST
```

VITA

Robert Thomas Kroutil

Candidate for the Degree of

Doctor of Philosophy

Thesis: PICOSECOND PHASE FLUOROMETRY

Major Field: Chemistry

Biographical:

Personal Data: Born in Oklahoma City, Oklahoma, July 6, 1955, the son of Mr. William R. Kroutil and Mrs. Beverly J. Kroutil.

Education: Graduated from Yukon High School, Yukon, Oklahoma, in May, 1973; received Bachelor of Science degree from Oklahoma State University in December, 1978; completed requirements for Doctor of Philosophy from Oklahoma State University in December, 1982.

Professional Experience: Teaching Assistant, Department of Chemistry, Oklahoma State University, 1978-1980; Research Assistant, Department of Chemistry, Oklahoma State University, 1981; IBM 370/168 Computer Operator, Computer Services, Oklahoma State University, 1982.