UNIVERSITY OF ILLINOIS

MAY 19 83

THIS IS TO CERTIFY THAT THE THESIS PREPARED UNDER MY SUPERVISION BY

SUSAN STEINKAMP SORLIE

ENTITLED PYROLYTIC PRODUCTION OF CARBON SELENIDE FROM CARBON DISELENIDE

IS APPROVED BY ME AS FULFILLING THIS PART OF THE REQUIREMENTS FOR THE

DEGREE OF BACHELOR OF SCIENCE IN CHEMISTRY

R. Linn Belford
Instructor in Charge

APPROVED:

HEAD OF DEPARTMENT OF CHEMISTRY

O-1364

PYROLYTIC PRODUCTION OF CARBON SELENIDE FROM CARBON DISELENIDE

BY

Susan Steinkamp Sorlie

THESIS

FOR THE

DEGREE OF BACHELOR OF SCIENCE

IN

CHEMISTRY

COLLEGE OF LIBERAL ARTS AND SCIENCES
UNIVERSITY OF ILLINOIS
URBANA, ILLINOIS

1983

TO PETE

ACKNOWLEDGEMENTS

I would like to thank Professor J. R. Marquart for his guidance and support this past year. His efforts have not only made this thesis possible but also made for an interesting and enjoyable learning experience.

I would also like to thank Dr. R. L. Belford and Nate Hennshaw for their friendship and support.

Finally I would like to thank my husband, Peter Sorlie, for postponing our move to California so I could finish this thesis.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

I. INTRODUCTION

This paper deals with the pyroltic production of CSe from carbon diselenide,

$$CSe_2 + Ar \rightarrow CSe + Se + Ar \ .$$

Interest in the decomposition of CSe_2 stems from previous studies of molecules isoelectronic with CSe_2 (CO_2, N_2O, CS_2, OCS) [1-4].

At this time little is known about CSe. In the past, investigations on CSe have been limited to collection of emission spectra and qualitative absorption data. For example, Callear and Tyerman produced CSe by flash photolysis of CSe_2 and observed its absorption spectrum [5]. However the spectrum did not show the relative intensities of the absorption peaks. Other studies have dealt with the emission spectra of CSe_2 and CSe [6-7] and the absorption spectrum of CSe_2 and Se [8-11]. To the author's knowledge this is the first quantitative study of CSe absorption.

An attempt was made first to obtain an absorption spectrum for CSe which included the relative intensities of the absorption peaks. Obtaining an absorption spectrum would have made a good starting point for an investigation of the properties of CSe. However, due to the lack of light intensity which resulted in a poor signal to noise ratio, poor resolution (broad spectral bandpass), the inability to reproduce shock conditions (the optical system could monitor only one wavelength per shock experiment) and the interference from CSe_2 hot bands, attainment of a spectrum proved impractical without modification of the experimental procedure.

The focus of this paper is on the determination of the rate constant, K(t), from a psuedo first order rate expression:

$$\frac{-\partial[CSe_2]}{\partial t} = K(t)[Ar][CSe] ,$$

where the concentration of argon is assumed to be time independent. The arrhenious parameters (pre-exponential factor and activation energy) are also evaluated. The results are compared with previous studies to help resolve the conflict arising from the two different theories which exist. These theories concern the mechanism by which the decomposition of CSe_2 proceeds. The first theory, developed by Graziano [11], states that the reaction proceeds via a two-step reaction:

$$CSe_2(^1\Sigma) + Ar \xrightarrow{K_1} CSe_2(^3B_2) + Ar$$

$$CSe_2(^3B_2) + Ar \xrightarrow{K_2} CSe(^1\Sigma) + Se(^3P_j) + Ar .$$

The second theory, developed by R. L. Belford and J. R. Marquart, states that the reaction occurs via a one-step process:

$$CSe_2(^1\Sigma) + Ar \xrightarrow{K} CSe(^1\Sigma) + Se(^3P_j) + Ar .$$

Since CSe is very unstable and short-lived (millisecond life-times), it had to be produced immediately before its spectrum is obtained. For this reason we produced it by pyrolysis of CSe_2 in a shock tube.

II. EXPERIMENTATION

The two inch in diameter shock tube used to thermally dissociate CSe_2 gas has been described in detail elsewhere [11], thus only a brief summary will be given here. Light produced by a deuterium lamp is passed laterally through the shock tube. It is then analyzed using a McPherson 218 grating monochromator and a RCA IP28A photomultiplier. To simplify data collection and analysis, a high speed transient recorder and an analog to digital converter is interfaced with a Vax/Vms-II computer. The system is triggered by a piezoelectric transducer which senses the approach of the shock wave. 2048 data points can be collected (at 0.5 microsecond intervals) for each shock. The experimental set up is shown in Figure 1. The shock tube also has interchangable incident and reflected endplates. The latter instantly reflects the shock wave back past the optical window while the former delays the reflected shock wave passage past the optical window by approximately 200 microseconds. The delay in the passage of the reflected wave allows for the absorption of light at the lower temperature T_2.* This absorption occurs after the incident shock wave passes the window, but before passage of the reflected shock wave which increases the temperature to T_5. This allows for the calculation of the absorbence of CSe_2 at temperatures where only a small amount of CSe_2 has been converted to products.

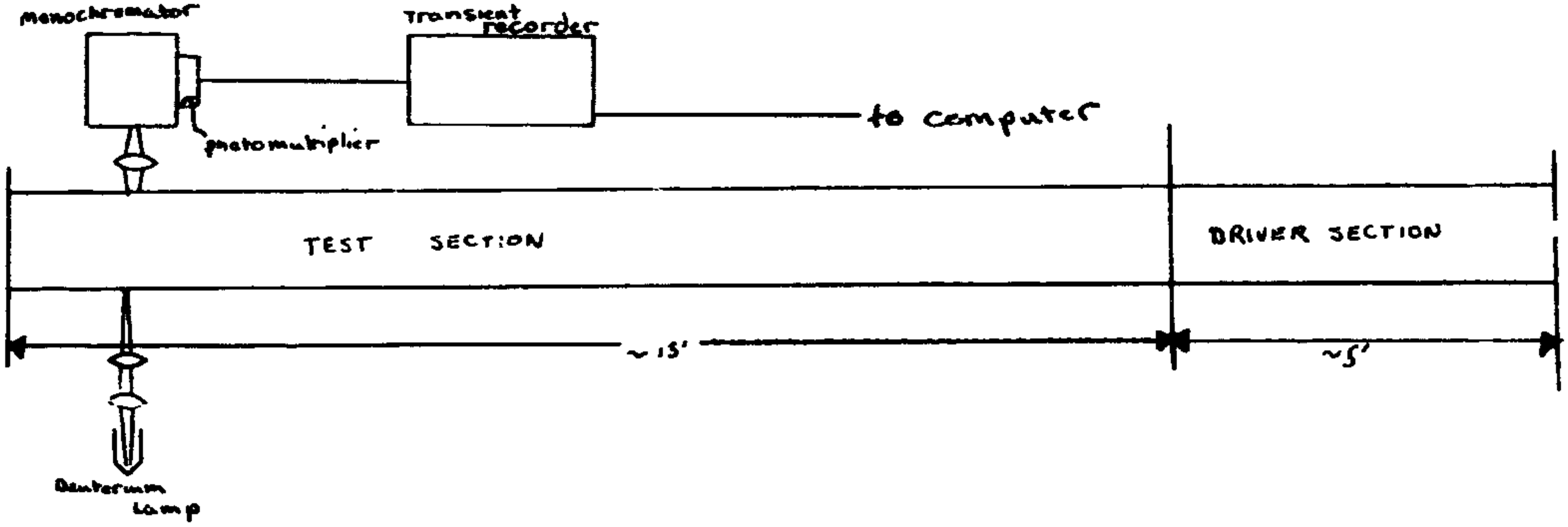
III. DATA ANALYSIS AND RESULTS

Light intensity data was collected and plotted against time for each shock run (see Figure 2). Figure 2a shows a shock run with the incident endplate and Figure 2b is with the reflected endplate. I_0 represents the incident light intensity. I_2, visible distinctly in Figure 2b, represents the emerging light intensity after the incident shock wave has passed. This halt is only momentary in Figure 2a. I_5^{∞} represents the intensity of light

---

*The subscript 0 refers to initial conditions, the subscript 2 refers to the conditions after the incident shock wave has passed, and subscript 5 refers to conditions after both shock waves have passed.

Figure 1. Experimental set up used for the collection of data in this work.


Monochromator
Transient recorder
photomultiplier
to computer
TEST SECTION
DRIVER SECTION
~15'
~5'
Deuterium lamp

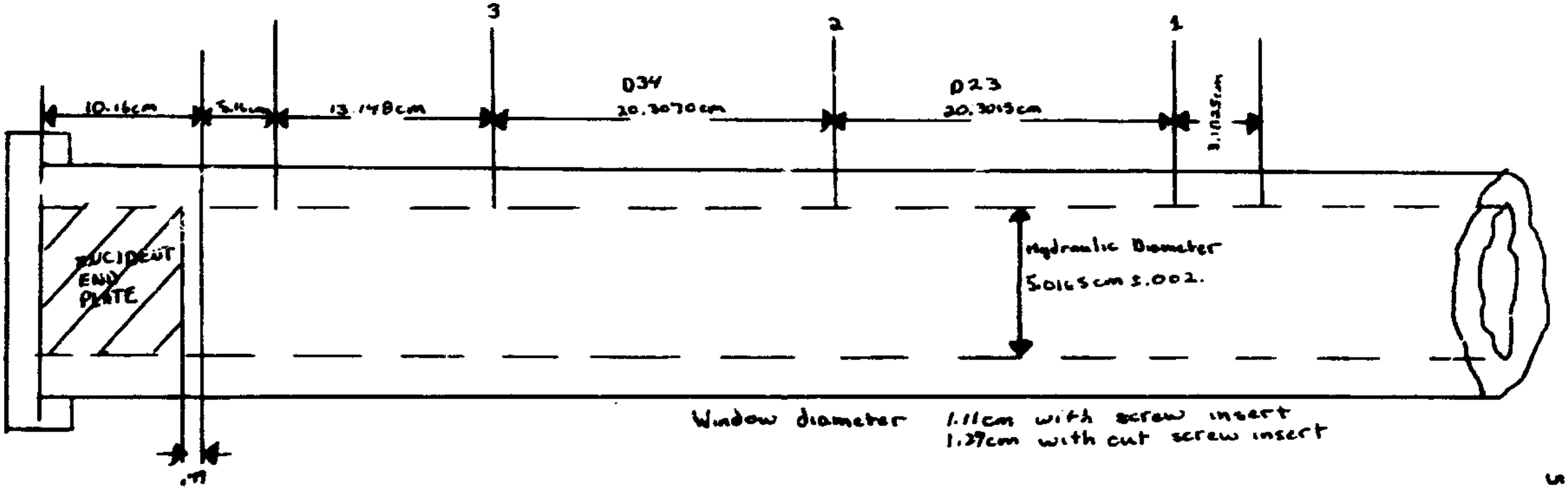
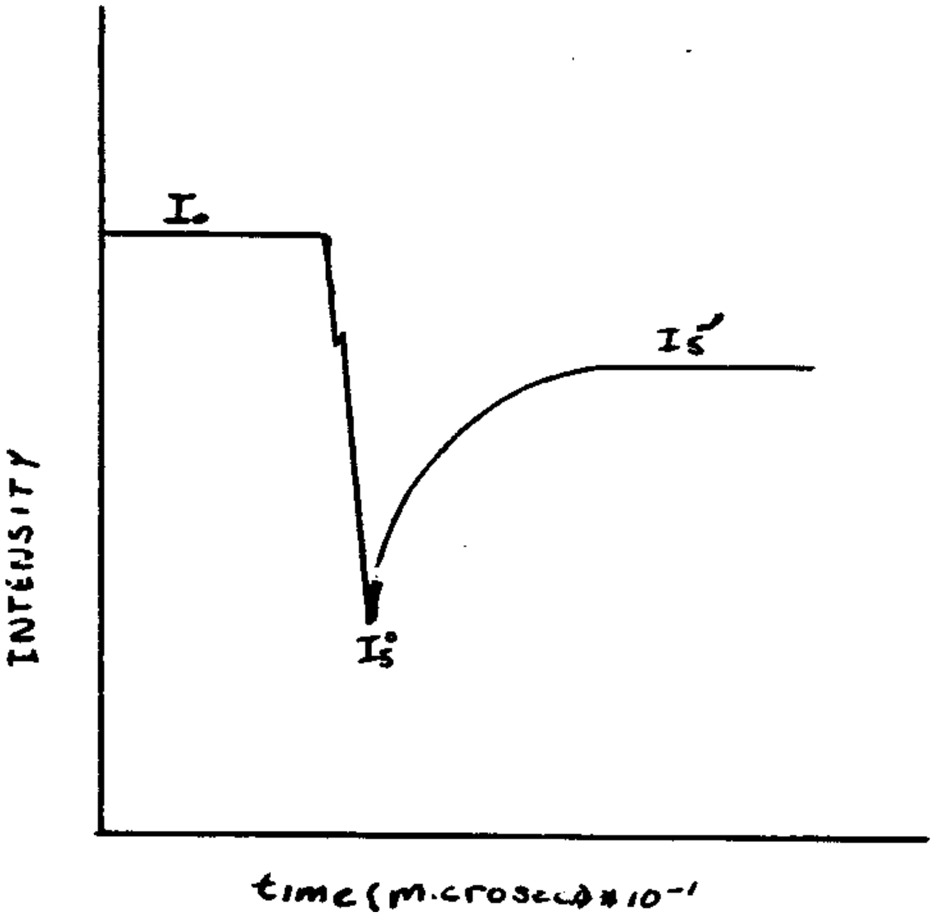
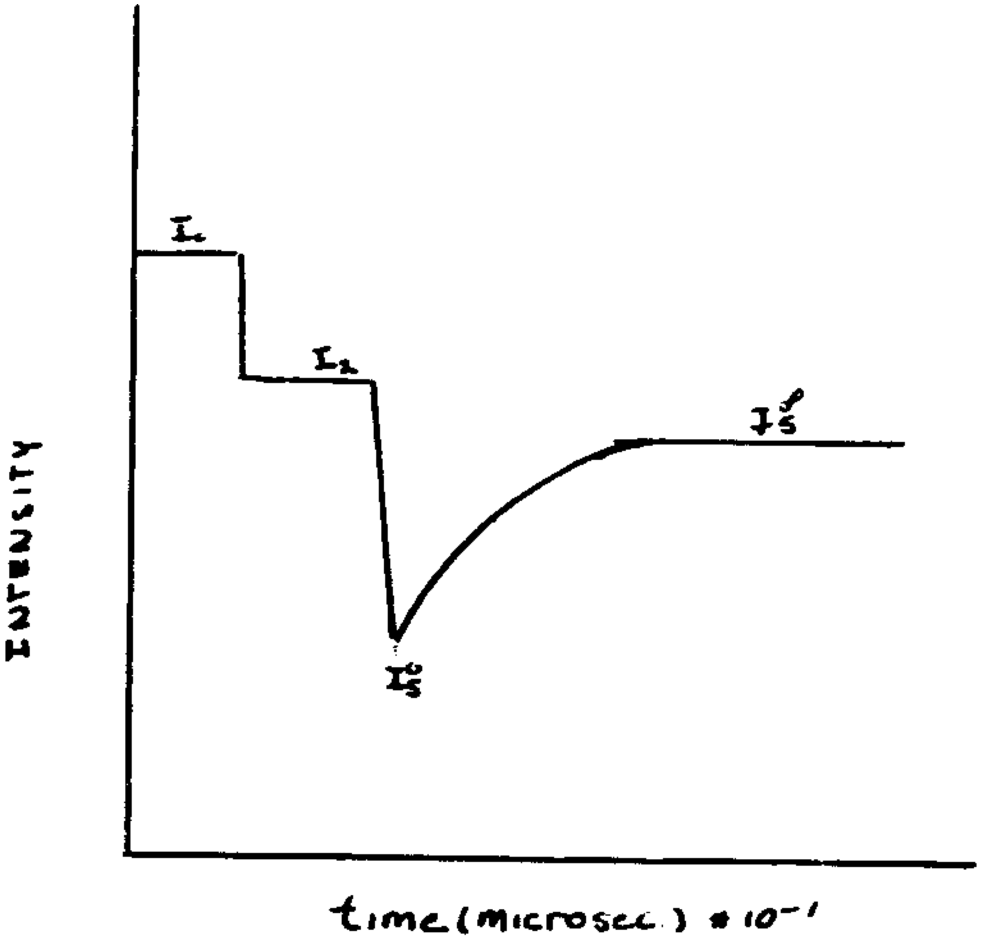

3
2
1
10.16cm
13.148cm
D34
20.3070cm
D23
20.3015cm
INCIDENT END PLATE
Hydraulic Diameter
5.0165cm ±.002.
Window diameter 1.11cm with screw insert
1.27cm with cut screw insert
.19

Figure 2. Graphs of intensity versus time. a) is with the reflected endplate and b) is with the incident endplate. I_0 is the incident intensity. I_2 (in b only) is the intensity after only the reflected wave has passed. I_5^0 is after both waves have passed but before any reaction has occurred and I_5^∞ is after both waves have passed and at equilibrium. (Graph drawn without noise for simplicity.)


I₀
Iₛ′
Iₛ⁰
INTENSITY
time(microsec)*10⁻¹

a


I₀
I₁
Iₛ′
Iₛ⁰
INTENSITY
time(microsec.) *10⁻¹

b

immediately after the incident and reflected waves have passed but before any CSe has been produced. And I_5^∞ is the light intensity after both waves have passed and the reaction has gone to completion or equilibrium. These graphs are drawn without noise for simplicity.

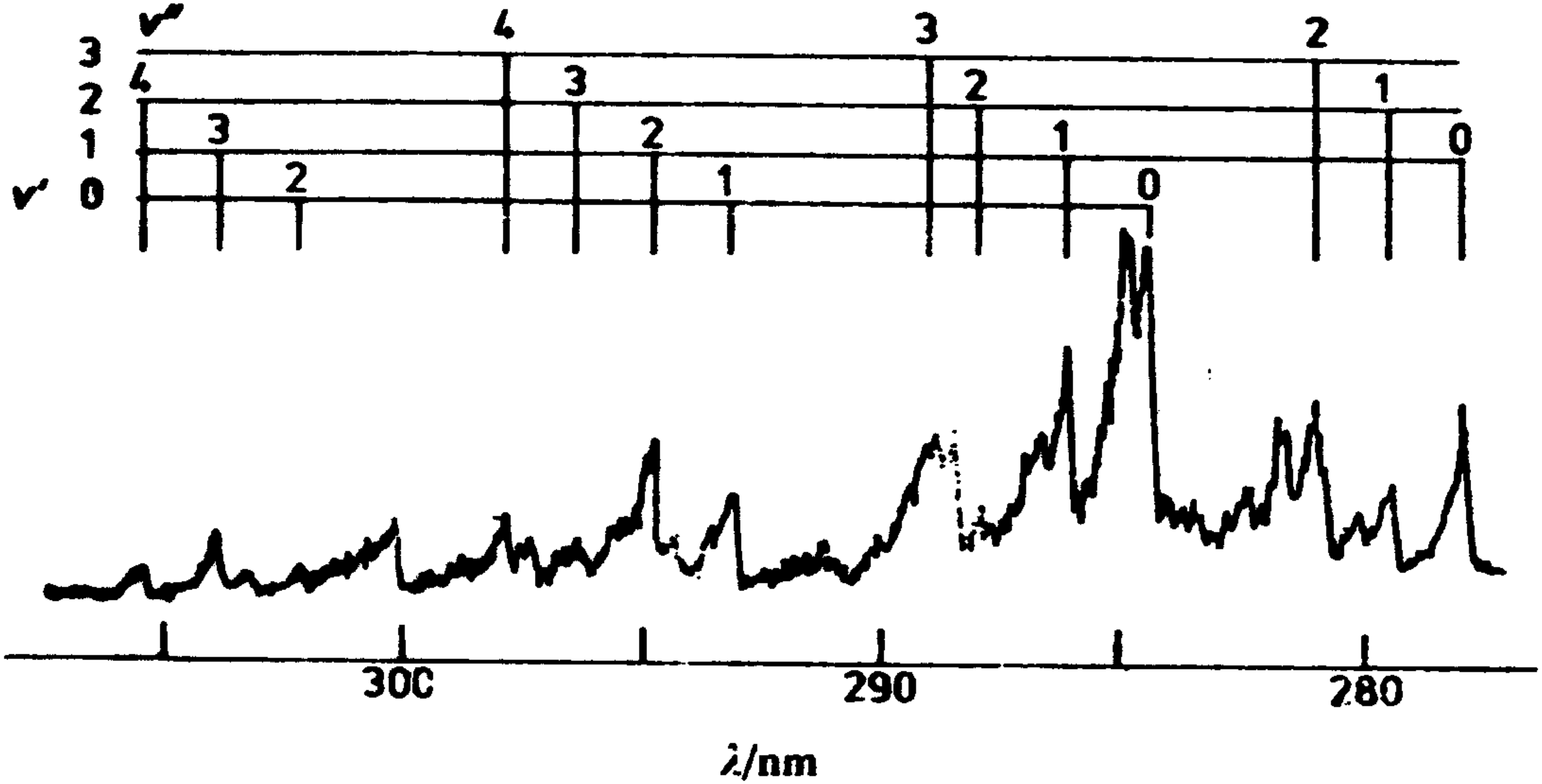
Alpha (see Appendix A) is a program written by J. R. Marquart to calculate the temperature, pressure, velocity, and density after the incident (T_2, P_2, V_2, ρ_2) and reflected (T_5, P_5, V_5, ρ_5) shock waves have each passed. These were calculated from the initial conditions (T_0, P_0, V_0, ρ_0) and the time required for the shock wave to pass two piezoelectric transducers spaced a known distance apart. The program was then modified to include calculations of the absorption coefficients (α_2, α_5^0, α_5^∞) and the fraction of CSe_2 converted to CSe, F, at infinite time (attainment of equilibrium) [8].

It should be mentioned that these calculations were done assuming ideal gas conditions. Boundary layer corrections or effects due to deceleration of the shock wave were not included. This does not effect data significantly when using the reflected endplate since deviations from ideal conditions are almost zero right after the incident wave has passed. The boundary corrections are significant when the incident endplate is in the shock tube since its reflected wave is delayed. The effect of these corrections on temperature is to increase T_2 by approximately eight degrees and to decrease T_5 by approximately two degrees.

Determination of the Rate Constant and Arrhenious Equation

The emission spectrum of CSe produced through photodissociation of CSe_2 at 147nm (see Figure 3) shows CSe's most intense transition is from the excited V' = 0 to the V" = 0 ground state (D → X) and occurs at

Figure 3. CSe (D → X) emission spectrum excited through photodissociation of CSe_2 at 147.0nm sample pressure, 0.05 torr, resolution 0.025nm [7].


v''
4
3
2
3
2
1
0
v'
4
3
2
1
3
2
1
0
2
1
0
300
290
280
λ/nm

approximately 2840 ang. [7]. For this part of the experiment, absorption measurements were made at 2846 ang. and a temperature range from 1700 K to 2200 K.

At 2846 ang., the absorption coefficients, α_2, α_5^0, α_5^∞ given in cm^2/mole, were found using the Beer-Lambert relationship:

$$\alpha_a^b = \ell n(I_0/I_a^b)/B\rho_a XCSe_2 \qquad (1)$$

where B = 5.0165 cm is the path length, ρ_a is the total gas density in moles/cc and $XCSe_2$ is the mole fraction of CSe_2 in argon, the diluent gas. This equation would give the absorption coefficient of CSe if all CSe_2 were pyrolyzed or of CSe_2 is none were pyrolysed. From studies on CSe_2, Marquart and Belford have found that at room temperature CSe_2 absorbs mainly at 2308 ang. [8]. At high temperatures, the CSe_2 absorption peak at 2308 ang. is broadened due to the formation of hot bands. This results in significant absorption at higher wavelengths. Thus in the temperature range, 1700 K to 2200 K, the absorption coefficient is equal to:

$$\alpha_a^b = (F)\alpha CSe + (1 - F)\alpha CSe_2 \qquad (2)$$

where αCSe_2 and αCSe are the absorption coefficients for CSe_2 and CSe respectively, and F is the fraction of CSe_2 converted to CSe.

After the incident shock wave has passed the temperature is around 1000 K. At temperatures less than 1200 K the amount of CSe produced is negligible (less than 0.1%), and the absorption coefficient calculated at T_2 is equal to the absorption coefficient of CSe_2

$$\alpha_2 = \alpha CSe_2 \qquad T < 1200 \text{ K} \ .$$

After the incident and reflected waves have passed but before any reaction has occurred, the absorption coefficient is equal to the CSe_2 absorption coefficient at T_5.

At temperatures greater than 2300 K only a negligible amount of CSe_2 exists (less than 0.8%) after a few microseconds. At these temperatures, the measured absorption coefficient is equal to the CSe absorption coefficient:

$$\alpha_5^{\infty} = \alpha CSe \qquad T > 2300\ K \quad .$$

Since most of the data collected was at temperatures between 1700 K and 2200 K it was necessary to determine the fraction of CSe_2 converted as a function of temperature.

At any time during the dissociation process the amount of CSe plus CSe_2 is a constant. Thus after both waves have passed

$$\rho_5 \chi CSe_2 = [CSe] + [CSe_2] \tag{3}$$

where ρ_5 is the density in moles/cc at T_5. The fraction of CSe produced is

$$F = [CSe]/\rho_5 \chi CSe_2 \quad . \tag{4}$$

The equilibrium constant (K_p) in cm^3/mole/sec for the reaction is

$$K_p = [CSe]^2 RT_5/[CSe_2] \tag{5}$$

where R is the gas constant. Solving (1) for the concentration of CSe_2, substituting into (3), solving for [CSe], and then plugging into (2), a relationship between the fraction and the equilibrium constant and temperature can be obtained:

$$F = \frac{1}{2}\left\{ -\frac{K_p}{RT} \pm \left[\left(\frac{K_p}{RT}\right)^2 + \frac{4K_p}{RT}\,\rho_5 \chi CSe_2\right]^{1/2} / \rho_5 \chi CSe_2 \right\} . \qquad (6)$$

The equilibrium constant in atmospheres as a function of temperature (Kelvin) has previously been determined by Marquart and Belford [8],

$$\log_{10} K_p = -.705 \log_{10}(T_5) + 7.64 \times 10^{-5}(T_5) + \frac{1.19 \times 10^4}{(T_5)^2} - \frac{1.85 \times 10^4}{T_5} + 9.68 .$$

Therefore, the fraction converted can now be determined as a function of temperature. The results of such calculations for each temperature is given in Table 1.

A plot of αCSe_2 versus temperature (see Figure 4) shows an exponential dependence of the CSe_2 absorption coefficient with temperature up to approximately 2200 K. Above this temperature the response of our electronic system was too slow to keep up with the fast change in intensity. This resulted in an experimental absorption coefficient that is probably lower than the true value. In the temperature range between 1700 K and 2200 K the exponential dependence can be approximated by a linear dependence without introducing too much error. The equation found is:

$$\alpha CSe_2 = 7143\ T - 6.875 \times 10^6 . \qquad (7)$$

Substituting (7) and (6) into (2) we can determine αCSe. The results are shown in Table 1. One can see that some of the values for the absorption coefficient were negative. This was due to some unexplained light emission found at high temperatures and at the larger of the two CSe_2 mole fractions used. As a result, these values were ignored. Upon inspecting the other

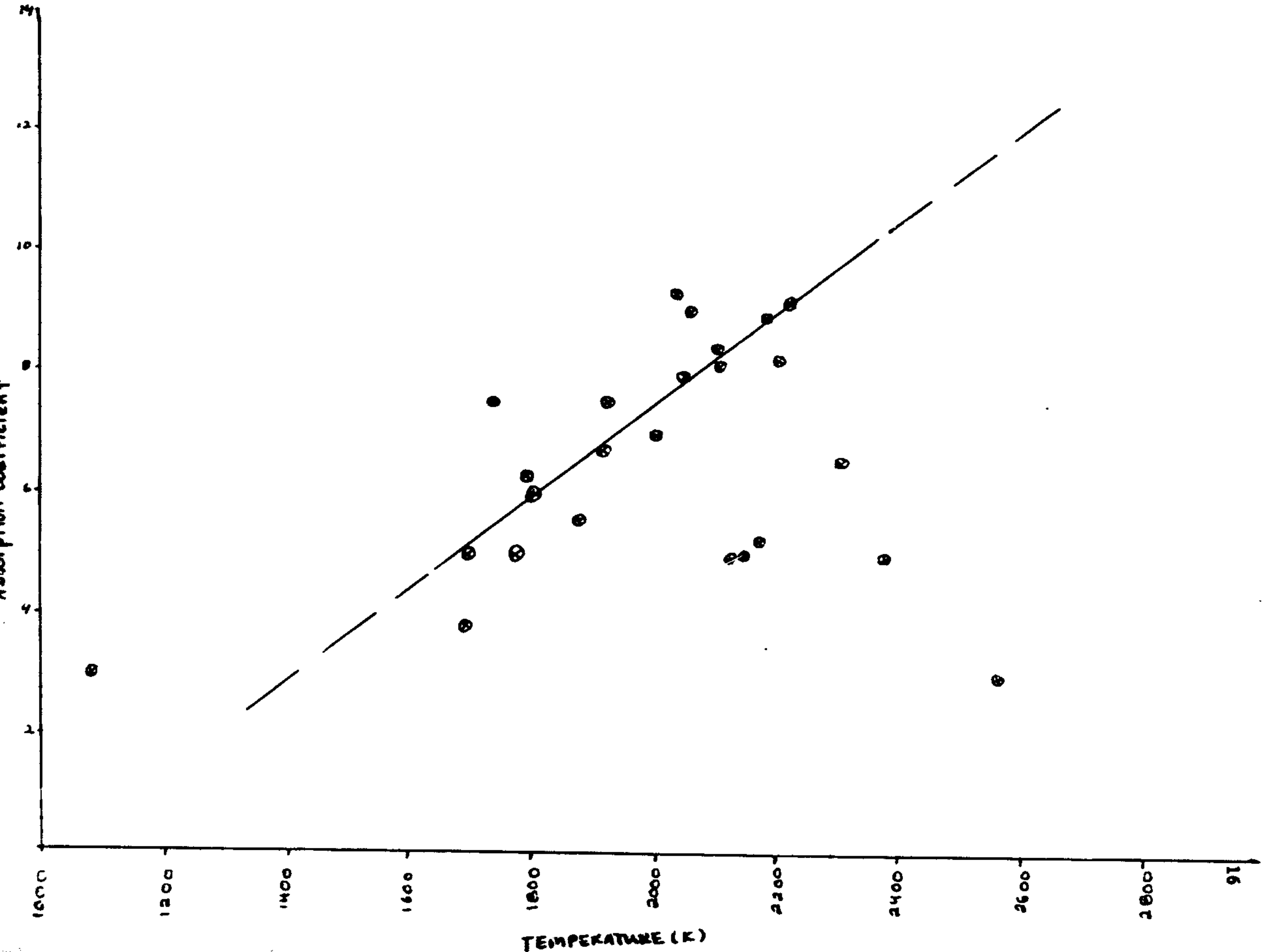
Table 1. Absorption Data

Endplate	SLIT	T_2	T_5	F	α_2	α_5^∞	$\alpha_5^0 = \alpha CSe_2$
Reflected	1/2	1187.8	2379.1	0.978	0	4.6837×10^5	5.0567 x 10
Reflected	1/2	1123.3	2224.4	0.936	0	7.5656×10^5	9.0133 x 10
Reflected	1/2	1051.6	2053.0	0.799	0	2.2774×10^6	9.1349 x 10
Reflected	1/2	1075.2	2109.7	0.860	0	1.6604×10^6	8.4085 x 10
Reflected	1/2	1059.2	2072.3	0.822	0	2.4253×10^6	9.3516 x 10
Reflected	1/2	1113.1	2201.1	0.925	0	1.5098×10^6	8.3503 x 10
Reflected	1/2	1083.1	2129.2	0.877	0	2.1576×10^6	8.6432 x 10
Reflected	1/2	1059.1	2069.6	0.820	0	3.3486×10^6	8.1036 x 10
Reflected	1/2	1108.7	2188.2	0.918	0	2.0587×10^6	9.1921 x 10
Reflected	1/2	976.2	1871.3	0.502	0	3.5159×10^6	5.5943 x 10
Reflected	1/2	936.8	1780.9	0.254	0	7.6358×10^5	6.2899 x 10
Reflected	1/2	962.4	1842.0	0.343	0	4.7627×10^5	5.9997 x 10
Reflected	1/2	1119.9	2218.4	0.905	0	9.2699×10^4	3.3747 x 10
Reflected	1/2	900.9	1695.1	0.139	0	1.0601×10^6	3.8493 x 10
Reflected	1/2	1092.4	2148.9	0.872	0	1.9672×10^5	4.7105 x 10
Reflected	1/2	1079.4	2118.0	0.785	0	2.3008×10^5	4.6812 x 10
Reflected	1/2	1142.4	2268.6	0.953	0	1.8994×10^6	6.4389 x 10
Incident	1	930.5	1764.3	0.582	4.8811×10^5	8.7897×10^6	5.1155 x 10
Incident	1	989.7	1905.7	0.835	1.2709×10^6	4.0683×10^6	7.9786 x 10
Incident	1	917.2	1734.8	0.516	1.0308×10^6	7.3282×10^6	7.5238 x 10
Incident	1	1050.1	2052.4	0.950	1.5052×10^6	3.7631×10^6	7.9507 x 10
Incident	1	1028.9	2001.9	0.924	2.9680×10^5	2.3250×10^6	7.1048 x 10
Incident	1/2	647.1	1089.9	.001	3.1306×10^5	9.6848×10^8	3.0694 x 10
Incident	1/2	1266.2	2567.5	0.999	3.1306×10^6	4.2262×10^5	3.1843 x 10
Reflected	1/2	1206.3	2425.6	0.997	0	2.4148×10^6	0
Reflected	1/2	1089.9	2146.8	0.977	0	2.7861×10^6	4.8205 x 10
Reflected	1	989.1	1903.8	0.832	0	2.8081×10^6	6.6902 x 10

Table 1. Absorption Data

SLIT	T_2	T_5	F	α_2	α_5^∞	$\alpha_5^0 = \alpha CSe_2$	αCSe
1/2	1187.8	2379.1	0.978	0	4.6837×10^5	5.0567×10^6	2.353×10^5
1/2	1123.3	2224.4	0.936	0	7.5656×10^5	9.0133×10^6	1.399×10^5
1/2	1051.6	2053.0	0.799	0	2.2774×10^6	9.1349×10^6	3.17×10^5
1/2	1075.2	2109.7	0.860	0	1.6604×10^6	8.4085×10^6	3.27×10^5
1/2	1059.2	2072.3	0.822	0	2.4253×10^6	9.3516×10^6	7.01×10^5
1/2	1113.1	2201.1	0.925	0	1.5098×10^6	8.3503×10^6	0.788×10^6
1/2	1083.1	2129.2	0.877	0	2.1576×10^6	8.6432×10^6	9.876×10^5
1/2	1059.1	2069.6	0.820	0	3.3486×10^6	8.1036×10^6	1.60×10^6
1/2	1108.7	2188.2	0.918	0	2.0587×10^6	9.1921×10^6	1.277×10^6
1/2	976.2	1871.3	0.502	0	3.5159×10^6	5.5943×10^6	-2.953×10^6
1/2	936.8	1780.9	0.254	0	7.6358×10^5	6.2899×10^5	-1.65×10^7
1/2	962.4	1842.0	0.343	0	4.7627×10^5	5.9997×10^5	0.491×10^6
1/2	1119.9	2218.4	0.905	0	9.2699×10^4	3.3747×10^5	-8.46×10^5
1/2	900.9	1695.1	0.139	0	1.0601×10^6	3.8493×10^5	-3.1369×10^7
1/2	1092.4	2148.9	0.872	0	1.9672×10^5	4.7105×10^5	-1.045×10^6
1/2	1079.4	2118.0	0.785	0	2.3008×10^5	4.6812×10^5	-2.037×10^6
1/2	1142.4	2268.6	0.953	0	1.8994×10^6	6.4389×10^6	1.438×10^6
1	930.5	1764.3	0.582	4.8811×10^5	8.7897×10^6	5.1155×10^6	4.66×10^6
1	989.7	1905.7	0.835	1.2709×10^6	4.0683×10^6	7.9786×10^6	2.73×10^6
1	917.2	1734.8	0.516	1.0308×10^6	7.3282×10^6	7.5238×10^6	2.14×10^6
1	1050.1	2052.4	0.950	1.5052×10^6	3.7631×10^6	7.9507×10^6	3.35×10^6
1	1028.9	2001.9	0.924	2.9680×10^5	2.3250×10^6	7.1048×10^6	1.71×10^6
1/2	647.1	1089.9	.001	3.1306×10^5	9.6848×10^8	3.0694×10^6	-1.27×10^8
1/2	1266.2	2567.5	0.999	3.1306×10^6	4.2262×10^5	3.1843×10^6	4.08×10^5
1/2	1206.3	2425.6	0.997	0	2.4148×10^6	0	2.38×10^6
1/2	1089.9	2146.8	0.977	0	2.7861×10^6	4.8205×10^6	2.588×10^6
1	989.1	1903.8	0.832	0	2.8081×10^6	6.6902×10^6	1.45×10^6

Figure 4. Absorption coefficient of CSe_2 versus temperature.


14
12
10
8
6
4
2
1000
1200
1400
1600
1800
2000
2200
2400
2600
2800
TEMPERATURE (K)

values obtained, CSe doesn't show any distinct temperature dependence in this temperature range. Therefore a statistical average for the absorption coefficient was calculated, giving a mean = 1.76 * 10**6 cm^2/mol and a standard deviation of 1.24 * 10**6 cm^2/mol.

Using the calculation of the absorption coefficient and other necessary data, the rate constant was evaluated.

The disappearance of $[CSe_2]$ can be described by the following differential equation:

$$\frac{\partial[CSe_2]}{\partial t} = -K(t)[Ar]\{[CSe_2] - [CSe_2]_{eq}\} \qquad (8)$$

where $[CSe_2]_{eq}$ is the equilibrium concentration of CSe_2. The fraction of CSe_2 not converted to CSe is:

$$1 - F = [CSe_2]_{eq}/\rho_5 \chi CSe_2 \ . \qquad (9)$$

The solution to the differential equation in (8) is:

$$[CSe_2] - (1 - F)\rho_5 \chi CSe_2 = B \exp(-K[Ar]t) \ . \qquad (11)$$

At T = 0,

$$B = [CSe_2]_0 - [CSe_2]_{eq} \ . \qquad (12)$$

Substituting (12) into (11) and solving for $[CSe_2]$ gives:

$$[CSe_2] = F[CSe_2]_0 \exp(-K[Ar]t) + (1 - F)[CSe_2]_0 \ . \qquad (13)$$

From Beer's law the absorption from the gas is

$$A = \{[CSe]\alpha CSe + [CSe_2]\alpha CSe_2\}B \qquad (14)$$

and

$$[CSe] = [CSe_2]_0 - [CSe_2] \; . \qquad (15)$$

Substituting (15) into (14), the absorpance is given as:

$$A = B[CSe_2]_0\alpha CSe + (\alpha CSe_2 - \alpha CSe)B[CSe_2] \; . \qquad (16)$$

Substituting the concentration of CSe_2 from (13) into (16) and setting all terms not dependent of time equal to A_5^∞ (16) becomes:

$$A - A^\infty = (\alpha CSe_2 - \alpha CSe)BF[CSe_2]_0 \exp(-K[Ar]t) \; .$$

Since the concentration of argon is constant and approximately equal to the desnity at T_5, K(t) becomes a psuedo-first order rate constant. Solving for K(t) in the above equation gives:

$$K(t) = \frac{-\ln(A - A^\infty) - \ln\{(\alpha CSe_2 - \alpha CSe)BF\rho_5\chi CSe_2]}{\rho_5 \cdot t} \; . \qquad (17)$$

Equation (17) and six of the graphs of intensity versus time were used to determine K(t) as a function of time during the reaction. The results are shown in Table 2. The values observed show that the rate constant remains constant within experimental error for the entire reaction. This supports the theory that the reaction proceeds via the Belford and Marquart one step mechanism [9] and not the Graziano two step mechanism [11]. If the latter were true, as the reaction proceeded, a decrease in the rate constant would be observed.

The value obtained for the rate constant should be considered a lower limit to the true value, since the data was not deconvoluted for time delay effects and it was not corrected for the boundary layer effects. A true

Table 2. Rate Constants (mole/cm^3/sec)

Time µsec \ Temp Kelvin	Run 1	Run 2	Run 3	Run 7	Run 5	Run 6
	1905.7	2001.9	2002.4	2053.0*	2072.3*	2109.7*
38.46				3.11×10^8	3.96×10^8	8.33×10^8
64	2.92×10^8	2.37×10^8	3.42×10^8			
76.7				5.58×10^9	5.02×10^2	8.83×10^8
96	2.69×10^8	2.06×10^8	3.01×10^8			
115.4				12.07×10^2	6.75×10^8	7.78×10^8
128	2.95×10^8	2.01×10^8	2.76×10^8			
160	2.93×10^8	2.08×10^8	2.68×10^8			
192	--	2.37×10^8	2.71×10^8			
224	--	2.29×10^8				
Average K(t)	2.87×10^8	2.20×10^8	2.92×10^8	6.92×10^8	5.24×10^8	8.31×10^8

*These runs were done at a higher concentration of CSe_2 and the light intensity data resulted in some unexplained emission which increased the observed K value making it too high.

graph of intensity versus time (see Figure 5) would probably descend deeper after the reflected shock has passed, and then as the reaction proceeds, it should probably rise faster. Equation (17) shows that both these convolution effects should cause a decrease in the value of the rate constant calculated, especially at higher temperatures. Boundary layer corrections would decrease αCSe_2 but not αCSe since it is apparently independent of small temperature changes. Again by (17) the resulting rate constant would decrease. These effects could be partially offset by the rather poor signal to noise ratio (approximately 5 for I_2) and a large monochromator slit width. Since the intensity of the dip was measured at the lowest possible point on the graph, any noise in the dip would cause the intensity to be smaller than the true value. Thus the calculated rate constant would be slightly larger than the actual value. A large slit width would also effect the rate constant. At 2846 ang., as at 2308 ang. in Figure 6, the absorption coefficient would be lowered by an increase in slit width. However since CSe_2 absorbance is due to hot bands, it varies slowly with wavelength, while αCSe probably varies more with wavelength in this range, so the lowering of the αCSe would be greater than that of αCSe_2. As a consequence the rate constant would decrease slightly.

Since so few points were obtained for the rate constant at different temperatures we could not accurately determine the arrhennius equation and thus the activation energy. Graziano determined the activation energy from an arrennius plot of $\ell n(K)$ vs. $1/T$ and it is shown in Figure 7 [11]. We added our data points to these (the circled X's in Figure 7). These points show good agreement with his low temperature points.

Figure 5. Graph of light intensity versus time. The dotted line shows the possible true value and the solid line shows the experimental values.

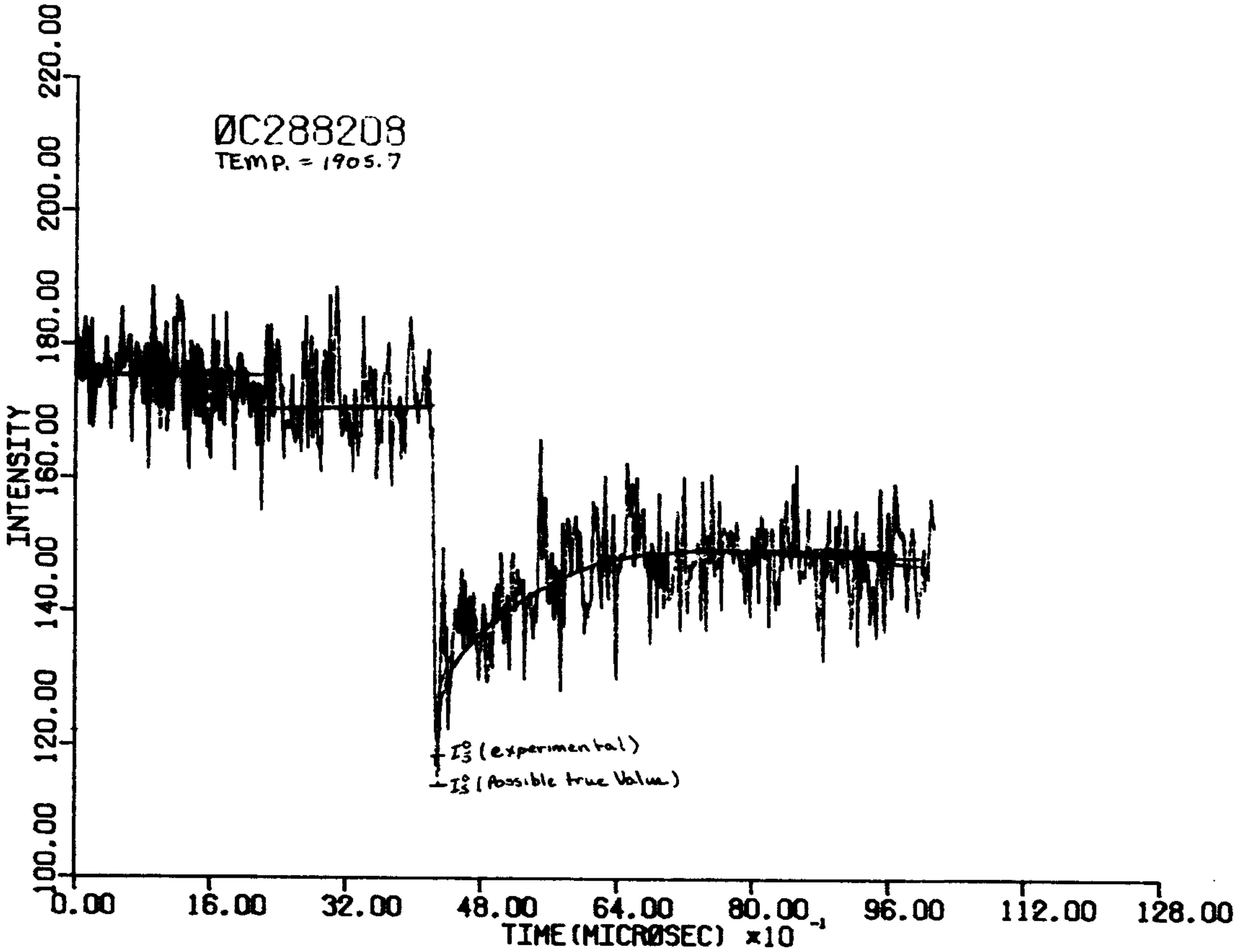

0C288208
TEMP. = 1905.7
I_s^0 (experimental)
I_s^0 (Possible true Value)
INTENSITY
100.00
120.00
140.00
160.00
180.00
200.00
220.00
TIME (MICR0SEC) ×10^{-1}
0.00
16.00
32.00
48.00
64.00
80.00
96.00
112.00
128.00

Figure 6. Absorption coefficient of CSe_2 versus slit width at room temperature and 2308 ang.

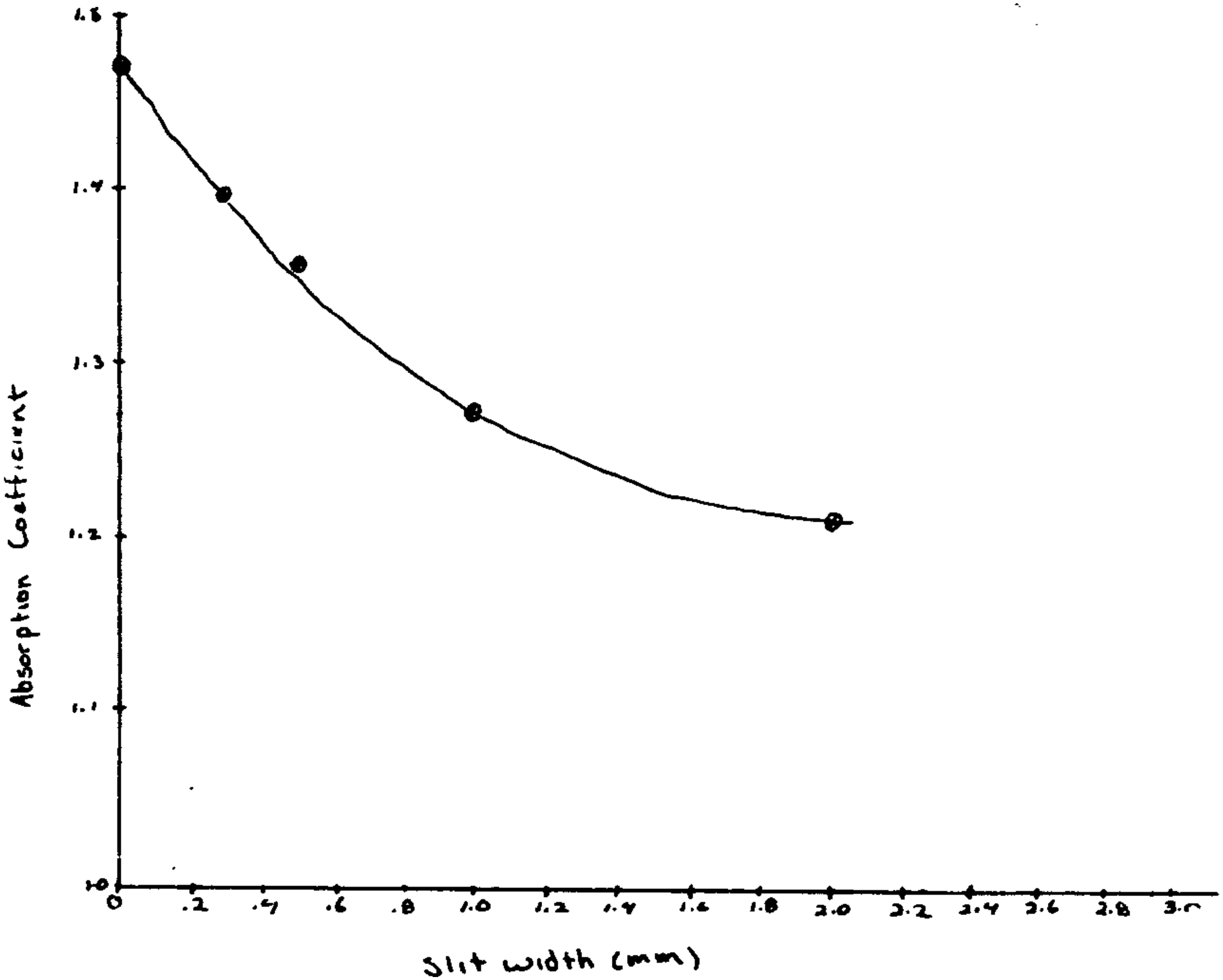
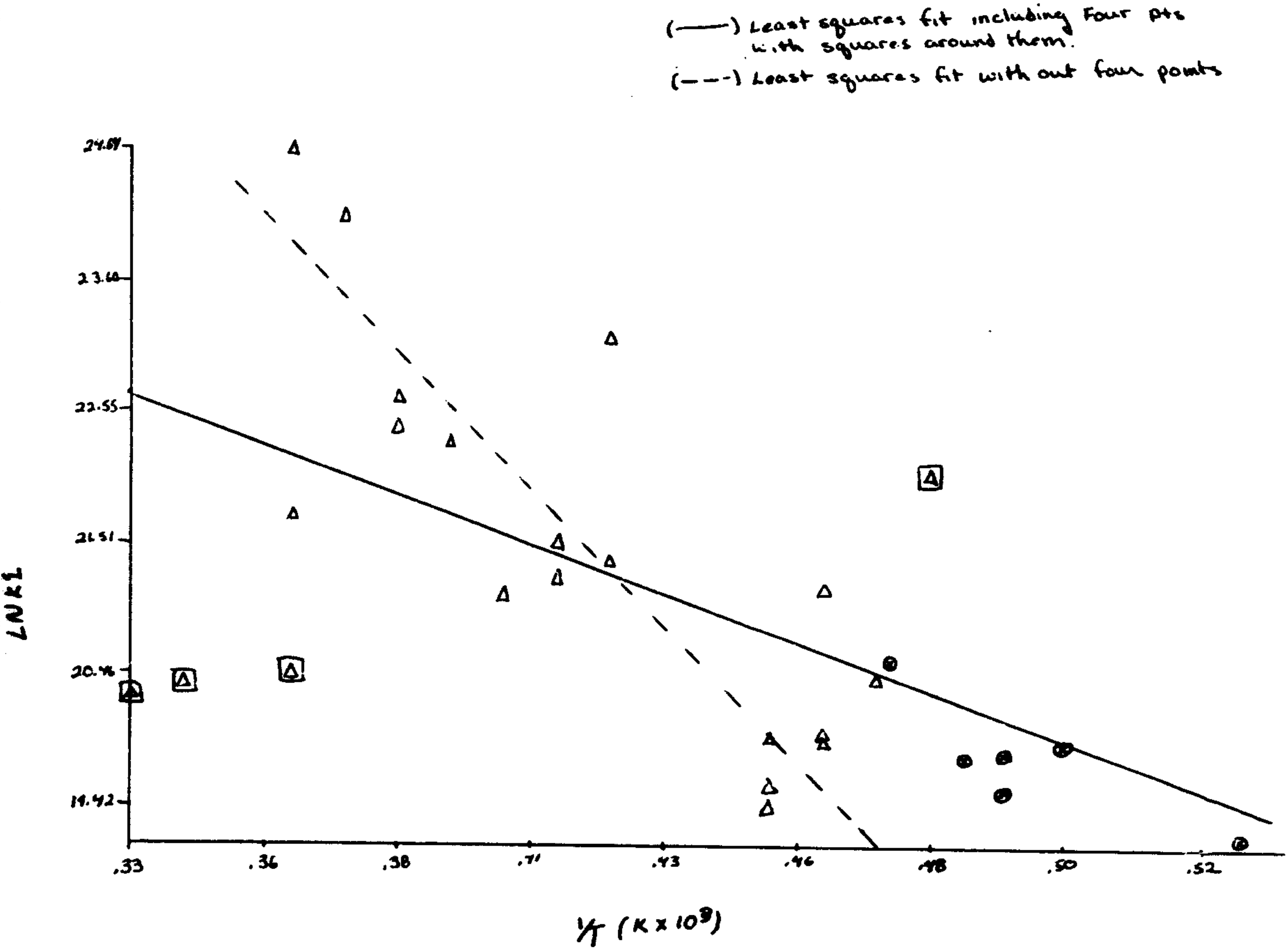

1.5
1.4
1.3
1.2
1.1
1.0
Absorption Coefficient
0
.2
.4
.6
.8
1.0
1.2
1.4
1.6
1.8
2.0
2.2
2.4
2.6
2.8
3.0
Slit width (mm)

Figure 7. Natural log of the rate constant versus inverse temperature. Δ are Graziano's points, $\otimes$ are our added points and $\boxed{\Delta}$ are Graziano's points excluded from our least squares fit.


(——) Least squares fit including four pts with squares around them.
(- - -) Least squares fit with out four points
24.69
23.60
22.55
21.51
20.46
19.42
LNK1
.33
.36
.38
.41
.43
.46
.48
.50
.52
1/T (K x 10^3)

Graziano did a least squares fit on his data points to extract the activation energy and obtained an Ea = 30.9 kcal/mole. After eliminating four of his points ($\boxed{\Delta}$) we obtained an activation energy of Ea = 62.2 kcal/mole. This value is in better agreement with a linear plot of the enthalpy of dissociation versus activation energy for CO_2, OCS NO_2, CS_2 and CSe_2 (see Figure 7). This plot was obtained by Belford and Marquart [9].

Absorption Spectrum for CSe

This section of the paper covers the technique used to find the absorption spectrum for CSe, the problems encountered, and possible alterations in experimental technique to alleviate these problems.

The data are obtained from 2700 ang. to 2900 ang. at temperatures ranging from 1600 K to 2500 K. The absorption coefficient was obtained as before using Beer's law for each wavelength.

DATAFIT (see Appendix B), a nonlinear least squares program, was used to fit a trial equation with an exponential dependence of the CSe_2 absorption coefficient on wavelength and $1/T_2$. The resulting equation is

$$\alpha CSe_2 = (.253 \times 10^9 - .170 \times 10^6 \; T_2)\exp(-1.36 \; \lambda/T) \; .$$

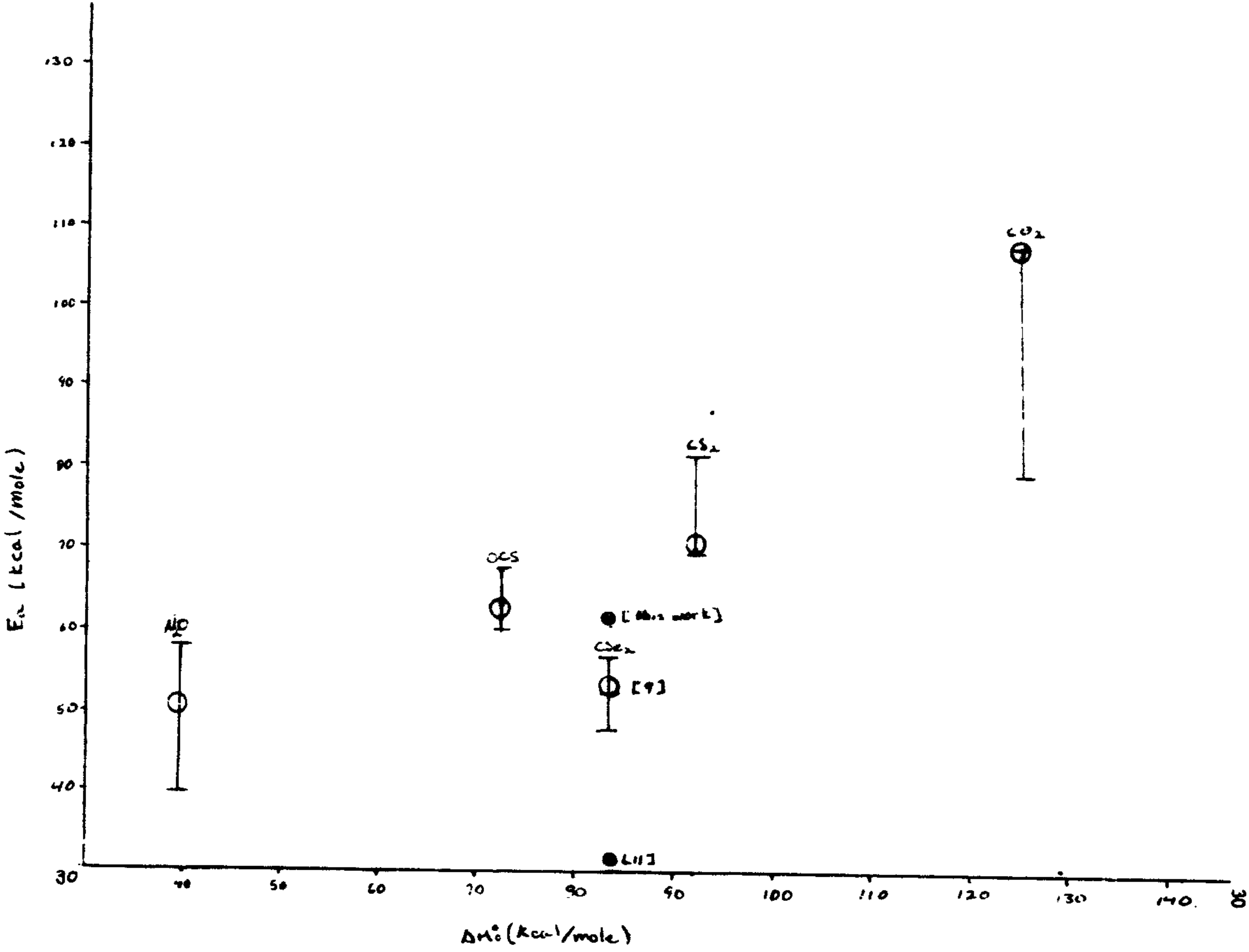
This equation was used to calculate the absorption coefficient of CSe_2 at T_5, however at 2000 K and 2846 ang., the calculated absorption coefficient is -1.25E07, an absurd negative result. It was concluded that this was not a good way of determining the absorption coefficient at T_5. It was later found in the calculation of the rate constant that the absorption coefficient for CSe_2 is a function of exp(T) and not exp(-1/T).

Another problem arose from the light source. Since the experiment was over in a few microseconds, it was necessary to open the slits to 1/2 or even 1 millimeter width. This resulted in poor resolution. Figure 8 shows the dependence of slit width on the absorption coefficient at 2308 ang. In the future the deuterium lamp should be replaced with a more intense light source to improve the signal to noise and allow for the use of narrower slits. (A DCL lamp produces light of greater intensity.)

Finally to determine the spectrum it would be best to be able to scan a number of wavelengths at a constant temperature. This is a problem which has been partially eliminated as we have become more accustomed to the shock tube and better able to reproduce shock conditions.

The modifications suggested above should make it possible to obtain an absorption spectrum in the future.

Figure 8. Enthalpy of dissociation versus activation energy for CO_2, CS_2, OCS, N_2O and CSe_2. The vertical bars represent the reported activation energies. The circles are typical of recently reported activation energies [9].


130
120
110
100
90
80
70
60
50
40
30
E_a (kcal/mole)
NO
OCS
CSe$_2$
CS$_2$
CO$_2$
[this work]
[9]
[11]
40
50
60
70
80
90
100
110
120
130
140
ΔH° (kcal/mole)

APPENDIX A

PROGRAM ALPHA

```
C       PROGRAM WRITTEN JANUARY,1983 BY J.R.MARQUART FOR THE
C       2 INCH SHOCK TUBE AT UNIV. OF ILLINOIS.
C       PROGRAM CALCULATES T2, P2, U2 ,T5, P5, U5 FOR A REAL GAS
C       CONSISTING OF NEARLY PURE ARGON. IT MAKES CORRECTIONS FOR
C       BOUNDARY LAYER EFFECTS.
C       IN THIS VERSION X2 = 1.67E-03
        REAL KP, T0, T2,T5,T25
        CHARACTER*30 CHAR
        OPEN(UNIT = 1,TYPE ='OLD',FORM = 'FORMATTED')
        OPEN(UNIT = 2,TYPE = 'NEW',FORM = 'FORMATTED')
        D23=20.3015
        D34=20.3070
        D4W=16.306
        DWR=0.09
        DWT=10.16
        DIA = 5.0165
        R1=8.3143E7
        R2=82.0575
        READ(1,70) NPTS
70      FORMAT(I2)
        DO 321 I = 1,NPTS
        READ(1,704) CHAR
704     FORMAT(A30)
        WRITE(2,705) CHAR
705     FORMAT(X,10('*'),A30,' ', 10('*'))
        READ(1,71)T1,P1
71      FORMAT(F4.1,X,F5.1)
        T1=T1+273.2
        P1=P1/760
        READ(1,72)T23,T34
72      FORMAT(2(F5.1,X))
        READ(1,73)IEP
73      FORMAT(I1)
C       THIS PART CALCULATES SHOCK VELOCITY AT END PLATE AND MACH
C       NUMBER
        DWEP=DWT
        IF(IEP.EQ.1) DWEP=DWR
        U23=1E6*D23/T23
        U34=1E6*D34/T34
        DECEL=(U23-U34)/((D23+D34)/2)
        UE=U34-DECEL*(D34/2+D4W+DWEP)
        U1=UE
        MW=39.948
        A=SQRT((1.6667*R1*T1)/39.948)
        AMACH=UE/A
        WRITE(2,110) AMACH
110     FORMAT(2X,'MACH = ',F7.4)
        D1=P1/(R2*T1)
        WRITE(2,105) T1,P1,D1
105     FORMAT(2X,'T1=',F5.1,4X,'P1=',F9.5,4X,'D1=',1PE12.4)
C       THIS PART SEARCHES FOR T2, P2, U2
        T2=T1+50.0
200     CONTINUE
        HT1=5.2033E6*T1-1.5515E9
210     CONTINUE
        HT2=5.2033E6*T2-1.5515E9
        U2=SQRT((HT1-HT2)*2+U1**2)
        T2CALC=T1*U2/U1+39.948*U2*U1/R1-39.948*U2**2/R1
        IF((T2CALC-T2).LE.0.1)GO TO 250
        T2=T2+0.1
        GO TO 210
```

```
250     CONTINUE
        D2=D1*U1/U2
        V2=U1-U2
        P2=U2*R2*T2
        WRITE(2,255)
255     FORMAT(2X,'WITHOUT BOUNDARY LAYER CORRECTIONS:')
        WRITE(2,257)
257     FORMAT(2X,'*********************************************
        WRITE(2,260) T2,P2,D2
260     FORMAT(2X,'T2=',F6.1,3X,'P2=',F10.5,4X,'D2=',1PE12.4)
        WRITE(2,270) U1,U2,V2
270     FORMAT(2X,'U1='1PE10.4,4X,'U2='1PE10.4,4X,'V2='1PE10.4)
C       THIS PART SEARCHES FOR T5, P5, D5
        T5=T2+50.0
300     CONTINUE
        HT5=5.2033E6*T5-1.5515E9
        U5=(HT5-HT2)/V2-V2/2
        W2=U5+V2
        T5CALC=T2*U5/W2+39.948*W2*U5/R1-39.948*U5**2/R1
        IF(ABS(T5CALC-T5).LE.0.1) GO TO 350
        T5=T5+0.1
        GO TO 300
350     CONTINUE
        D5=D2*W2/U5
        P5=U5*R2*T5
        WRITE(2,400) T5,P5,D5
400     FORMAT(2X,'T5=',F6.1,3X,'P5=',F10.5,4X,'D5=',1PE12.4)
        WRITE(2,450) U5,W2
450     FORMAT(2X,'U5=',1PE10.4,4X,'W2=',1PE10.4)
C       THIS PART OF THE PROGRAM CALC ROUGHLY WHERE THE
C       REF. WAVE  SHOULD APPEAR.
        IF (IEP .EQ. 0) THEN
        TIME2 = 10.306/U1
        TIME5 = TIME2 + DWT/U5 + DWI/U1
        WRITE(2,3)
3       FORMAT(' ', ' THE REF. AND INC. WAVES  PASS THE WINDOW AT:'
        WRITE(2,4) TIME5, TIME2
4       FORMAT(' ', 'TIME5= ',F9.4,'     TIME2= ', F9.4)
        ENDIF
C       THIS PART OF THE PROGRAM IS DESIGNED TO CALCULATE THE
C       ABSORPTION COEF. FOR CSF  AND CSF2 USING BEER'S  LAW AND
C       DATA ABOVE W/OUT BOUNDARY LAYER CONDITIONS.  BASE E
        X2 =1.67E-03
        RCM = 5.0165
        R = 82.06
        C2 = D2*X2
        READ(1,74)  T0, T2,  T5
74      FORMAT(3(F3.0,X))
C       CALCULATE THE FRACT. OF CSF2 CONVERTED TO CSF (F)
        AN= -0.705*ALOG10(T5) + 7.64E-05*T5 + 1.19E+04/T5**2
        BN = -1.85E+04/T5 + 9.66
        AN = AN + BN
        KP = 10**AN
        G = KP/(R*T5)
        C5  = 0.5*(-G + SQRT(G**2 + 4*G*D5*X2))
        C3 = X2*D5
        F =  C5/(D5*X2)
        WRITE(2,5) F
5       FORMAT(2X, 'THE FRACTION OF CSF2 CONVERTED TO CSF IS =',F7.
C       CALC. THE ABSORB. COEFF. (A2 AND A5)
        READ (1,76) WAVENU
```

```
76      FORMAT(F6.1)
        IF (WAVENU .EQ. 2306.0) THEN
        C1 = X2 * D1
        C5 = (1-F)*D5*X2
        I1 = I0
        READ (1,*) SLIT
                IF (SLIT .EQ. 1.0) THEN
                A1 = 1.27E 06
                ELSEIF (SLIT .EQ. 0.5) THEN
                A1 =  1.355E 06
                ELSE IF (SLIT .EQ. 0.3) THEN
                A1 = 1.395E 06
                ELSE
                WRITE(2,6)
6       FORMAT( ' VALUES FOR THAT SLIT WIDTH HAVEN''T BEEN CALC.')
                ENDIF
        A2 = (ALOG(I1/T2)/BCM + A1*C1)/C2
        A5 = (ALOG(I1/T5)/BCM + A1*C1)/C5
C       THE  CALC FOR THE OLD VALUE OF A2 AT T2
        AOLD = 2.230E 06 + 7.167E 10/T2 - 1.083E 13/T2**2
        WRITE(2,7) AOLD
7       FORMAT(' ', 'THE OLD VALUE FOR A2 AT T2 IS ', 1PE12.4)
        ELSE
        A2 = (ALOG(I0/T2))/(BCM*C2)
        A5 = (ALOG(I0/T5))/(BCM*C5)
        ENDIF
        IF (F .LT. .10) THEN
        AOLD5 = 2.230E 06 + 7.167E 10/T5 - 1.083E 13/T5**2
        WRITE(2,8) AOLD5
8       FORMAT(' THE OLD VALUE FOR A2 AT T5 IS ' , 1PE12.4)
        END IF
        FACTOR = 1
        IF (SLIT .EQ. .3)  FACTOR = .867
        IF (SLIT .EQ. .5) FACTOR =  .893
        IF (SLIT .EQ. 1.0) FACTOR = .953
        A2 = A2 * FACTOR
        A5 = A5 * FACTOR
        WRITE(2,9)
9       FORMAT(' THE ABSORB. COEF. IS GIVEN WITH A 2NM SLITIF AT 23
        WRITE(2,11)  A2
        WRITE(2,444)
        WRITE(2,12) A5
11      FORMAT(2X,'THE ABSORB. COEF. FOR CSE2 IS',1PE12.4,/)
12      FORMAT(2X,'THE ABSORB. COEF. DIV. BY IS',1PE12.4,/)
        READ(1,77) I25
77      FORMAT(F4.0)
        C25 = D5 * X2
        A25 = (ALOG(I0/I25))/(BCM*C25)
        WRITE (2,444)
444     FORMAT( ' AT I5')
        WRITE(2,11) A25
        RA5 = A5 - (7103*I5 - 665/000)*(1-F)/F
        WRITE(2,454) RA5, T5
454     FORMAT(' THE ABSORB. COEF OF CSE IS ',1PE12.4,' AT ',1PE12.4
        RA55 = A5 - A25*(1-F)/F
        WRITE(2,453) RA55, I5
453     FORMAT(' THE ABSORB COEFF OF CSE IS ',1PE12.4,' AT ',1PE12.4)
321     CONTINUE
        WRITE(2,500)
500     FORMAT(2X,'WITH BOUNDARY LAYER CORRECITONS')
        WRITE(2,550)
```

```
550     FORMAT(2X,'***************************************************
        TL=DWEP*DS/(D1*U1)+DWEP/U5
        TH=4.7836*DIA**2*P1*760/((AMACH**2.07)*A)
        R=SQRT(TL/TH)
        T2=T2*(1+0.2*R)
        D2=D2*(1.0+0.2*R/(1.6667-1.0))
        P2=D2*R2*T2
        WRITE(2,600) T2,P2,D2
600     FORMAT(2X,'T2=',F6.1,3X,'P2=',F10.5,4X,'D2=',1PE12.4)
        TS=T2+50.0
650     CONTINUE
        HTS=5.2033E6*TS-1.5515E9
        US=(HTS-HT2)/V2-V2/2
        W2=US+V2
        TSCALC=T2*US/W2+39.948*W2*US/K1-39.948*US**2/K1
        IF (ABS(TSCALC-TS).LE.0.1) GO TO 700
        TS=TS+0.1
        GO TO 650
700     CONTINUE
        DS=D2*W2/US
        PS=DS*R2*TS
        WRITE(2,750) TS,PS,DS
750     FORMAT(2X,'TS=',F6.1,3X,'PS=',F10.5,4X,'DS=',1PE12.4)
        WRITE(2,800) US,W2
800     FORMAT(2X,'US=',1PE10.4,4X,'W2=',1PE10.4)
        STOP
        END
```

APPENDIX B

PROGRAM DATA FIT

```
      PROGRAM DATAFIT
C     PROGRAM DATAFIT IS FROM RONALD FOSNER'S THESIS, 1981
C                      AND WAS EDITED FOR VAX BY SUSAN SORLIE, 1983
C     DATAFIT IS A NONLINEAR LEAST SQUARES ALGORITHM THAT CAN
C     BE USED TO SOLVE ANY DIFFERENTIABLE NON-LINEAR EQUATION WITH 30
C     OR LESS PARAMETERS AND 1000 OR LESS DATA SETS.  DATAFIT USES THE
C     MARQUARDT ALGORITHM TO OBTAIN A COMPROMISE BETWEEN THE GRADIENT
C     METHOD AND THE LINEARIZATION METHOD.
C
C     THE USER HAS THE OPTION OF CHOOSING ADDITIONAL OUTPUT TO ACCOMPANY
C     THE LISTING OF LEAST SQUARES VALUES FOR THE MODEL PARAMETERS
C     AND THEIR ESTIMATED STANDARD ERRORS. THIS ADDITIONAL OUTPUT IS
C     INTENDED TO HELP THE USER DETECT IF THE LOCAATER MINIMUM OF THE
C     LEAST SQUARES HYPERSURFACE THAT THE PROGRAM HAS LOCATED IS IN
C     FACT THE LOCATION OF THE DESIRED GLOBAL MINIMUM.
C
C     THE USER MUST SUPPLY A FUNCTION SUBROUTINE ENTITLED -FUNCTN-
C     THAT, WHEN SUPPLIED WITH THE CORRECT VALUE OF THE INDEPENDANT
C     VARIABLES AND CURRENT MODEL PARAMETERS WILL GIVE THE VALUE
C     OF THE DEPENDANT VARIABLE.
C     SEE THE  SAMPLE FUNCTN SUBROUTINE SUPPLIED AT THE END OF  THIS
C     PROGRAM.
C
C     THE USER MUST ALSO SUPPLY THE PROGRAM WITH A DATA FILE THAT
C     CONTAINS THE DATA, INITIAL PARAMETER GUESSES, DELTA B VALUES,
C     PARAMETER NAMES, ETC. THAT ARE USED FOR COMPUTATION AND OUTPUT.
C
C     THE INPUT FILE MUST CONTAIN THE FOLLOWING DATA IN THIS ORDER:
C
C     1. A TITLE OF THE RUN 12 CHAR MAX
C     2. # OF PARAMETERS, # OF INDEPENDENT VARIABLES, # OF DATA SETS
C        10 INDEPENDENT VARIABLES MAX.
C     3. EACH PARAMETER NAME FOLLOWED BY THE INITIAL GUESS OF THAT
C        PARAMETER AND THE VALUE OF DELTA B TO BE ADDED TO COMPUTE
C        THE DERIVATIVE BY FINITE DIFFERENCES.  THEREFORE 2 CARDS
C        FOR EACH PARAMETER. (E11.5, 2X, F6.1)
C     4. A CARD CONTAINING THE DEPENDENT VARIABLE NAME 12 CHAR MAX
C     5. CARDS CONTAINING EACH OF THE DEPENDENT VARIABLE'S NAMES
C        12 CHAR MAX
C     6. THE # OF DATA POINTS TO BE DELETED BETWEEN 0 AND 30
C     7. A CARD FOR THE DATA POINTS TO BE DELETED WHICH CONTAINS THE 4
C        CHAR RUN NAME OF THAT DATA POINT. (IN INPUT ORDER)
C     8. THE OPTIONS WHICH CAN BE:
C           SEE  - TO SEE EACH ITERATION OF THE PROGRAM
C           SEE+ - OPTION SEE PLUS SHOWING EACH MATRIX MANIPULATION.
C           LRES - PRINT RESIDUALS IN A LIST
C           PRES - PLOT OUT THE RESIDUALS IN PLOTS
C           LAMB - RESET THE VALUE OF LAMBDA: A VALUE BETWEEN 1000 AND
C                  .001 MUST FOLLOW THIS CARD. (F8.3)
C           LIMI - RESET THE # OF DECIMAL OF NO CHANGE THAT THE SUM
C                  OF SQUARES IS TO REACH IN ORDER TO TERMINATE THE
C                  PROGRAM: 5 IS DEFAULT AND A VALUE BETWEEN 1 AND
C                  14 MUST FOLLOW. (I4)
C           LOOP - THE MAX ITERATIONS: DEFAULT 30 (I4)
C           REST - RESET VALUE OF LAMBDA IF NO CONVERGENCE AFTER MAX ITER
C                  DEFAULT IS 100,000. (F8.3)
C           SMRP - UNCOMPLETED DO NOT USE
C           STOP - THIS MUST BE THE LAST OPTION AND MUST BE INCLUDED
C                  EVEN IF NO OTHER OPTIONS ARE CHOSEN.
C    9. THE DATA SET CARDS:
C       EACH CARD CONTAINS THE 4 CHARACTER RUN LIST THAT IS ANY
```

```
C       ARBITRARY TAG THAT THE USER CARES TO PLACE ON EACH DATA SET.
C       THE VALUE OF THE DEPENDENT VARAIBLE FOLLOWED BY EACH OF THE
C       DEPENDENT VARIABLES IN ORDER FROM X(N,1) TO X(N,I).  IF THERE
C       ARE I INDEPENDENT VARIABLES AND N DATA SETS.
C       THERE MUST BE A CARD FOR EACH SET OF POINTS IN THE DATA SET.
C
C          A SAMPLE INPUT FILE IS SHOWN BELOW:
C
CGAUSSIAN AND QUADRATIC PLUS RANDOM ERROR
C 2X 1Y  30
CMULTIPIPLIER
C.21000E+03XX    2.0
CB2
C.56500E+03XX    1.0
CDEPENDANT
C COUNT
C    0
CPRES
CLIMT
C   10
CSTOP
C  'D   1'XXX.1979E+03XX.50000
C  'D   2'   A NEW FORMAT STATEMENT IS NEEDED
C  'D   4'   FOR MOST DATA SETS (SEE LABEL 16 IN MAIN PROGRAM)
C
C   NOTE: IF THE DATA SET IS LARGE, USE ABOUT 100 POINTS TO GET
C         AN INITIAL GUESS FOR THE PARAMETERS FOR THE FULL SET.
C         THIS WILL SPEED THINGS UP.
C   ************************
C
C

        COMMON/NAME/BNAME
        COMMON/INSTUFF/ NIV, NTERMS, NPTS,NDEL,B, DELTAB,X,Y
        COMMON/RESULT/ YCALC, SUMSQR, SIGMAB, ITER,  LAMBDA
        COMMON/OPTIONS/  CLAM, SLIM, SOUP, RLAM
        COMMON/OPT/OPT
        CHARACTER*4 OPTION(10), O
        REAL YCALC(1000), SUMSQR, SIGMAB(30), LAMBDA
        INTEGER TITLE(9), HEAD(9)
        CHARACTER*4 BNAME(30,3), XNAME(10,3),YNAME(3), DNAME(1000)
        CHARACTER*4 NDELETE(30)
        INTEGER SOUP, SLIM
        REAL Y(1000), X(1000,10), B(30), DELTAB(30) ,CLAM
        INTEGER ORDER(1000), OPT(10)
        DATA HEAD/ 'NON', 'LINE', 'AR L', 'EAST', ' SQU', 'ARES',
     1            ' DAT', 'A FI', 'T VA'/
        DATA OPTION/ 'SEE ', 'CRES', 'PRES', 'REST', 'LAMB',
     1               'LIMT', 'LOOP', 'SMRP', 'SEE+', 'STOP'/
        DATA OPT/10*0/
        OPEN(UNIT = 22, TYPE = 'OLD', FORM = 'FORMATTED')
        OPEN(UNIT = 19, TYPE = 'NEW', FORM = 'FORMATTED')
C
C   READ IN DATA AND OPTIONS
C
        READ (22, 1050), (TITLE(J),J =1,9)
        READ(22,13), NTERMS, NIV, NPTS
13      FORMAT (I2,X,I2,X,I4)
        DO 31 K = 1, NTERMS
        READ (22,1050), (BNAME(K,I), I =1,3)
31     READ(22,14), B(K), DELTA B(K)
14      FORMAT( 2X,F5.2,X,F5.0)
```

```
      READ (22,1050), (YNAME(I), I =1,3)
      DO 1044 J = 1,NIV
1044  READ (22,1050), (XNAME(J,I), I =1,3)
      READ(22,15), NDEL
15    FORMAT(I4)
      IF(NDEL .EQ. 0) GOTO 12
      READ (22,1054), (NDELETE(I), I = 1,NDEL)
1054  FORMAT(30(A4,X))
12    DO 3 M1 = 1,15
      READ (22,1052), D
      IF (D .EQ. 'STOP') GOTO 4
      IF (D .EQ. 'LAMB') READ(22,18), CLAM
      IF (D .EQ. 'LIMI') READ(22,17), SLIM
      IF (D .EQ. 'LOOP') READ(22,17), SLOP
      IF (D .EQ. 'REST') READ(22,18), RLAM
      DO 2 M = 1,10
2     IF (D .EQ. OPTION(M)) OPT(M) = 1
17    FORMAT(I4)
18    FORMAT(F8.3)
3     CONTINUE
4     DO 9 M2 = 1,NPTS
      READ(22,16), DNAME(M2), Y(M2), (X(M2,J), J = 1,NIV)
      WRITE(19,16),DNAME(M2), Y(M2), (X(M2,J), J=1,NIV)
16    FORMAT(X,A4,X,F5.2,X,F8.3)
9     CONTINUE
      IF (NDEL .EQ. 0) GOTO 25
      DO 25 M2 =1,NPTS
      DO 10 N2 = 1,NDEL
      IF (DNAME(M2) .EQ. NDELETE(N2)) THEN
      NPTS = NPTS - 1
      DO 30 I = M2,NPTS
       DO 20 J = 1,NIV
       X(I,J) =X(I+1,J)
20     CONTINUE
      Y(I) = Y(I+1)
      DNAME(I) = DNAME(I+1)
30    CONTINUE
      ENDIF
10    IF (DNAME(M2+1) .EQ. NDELETE(N2)) GOTO 25
25    CONTINUE
1050  FORMAT(33A4)
1052  FORMAT(A4)
      CALL  HEADING (TITLE, HEAD)
      CALL INOUT
      WRITE (19, 1060)
1060  FORMAT ('0', 110('*'))
      CALL FITTER
      CALL OUTOUT
      IF (OPT(2) .EQ. 1) CALL RESID(Y, YCALC, DNAME, TITLE, NPTS)
      IF (NIV .EQ. 1) THEN
      CALL SIGPAR(SUMSQR,X,NPTS)
      ENDIF
      IF(OPT(3) .EQ. 1 .OR. OPT(8) .EQ. 1) THEN
      CALL RESTOP(Y, X, NIV, YCALC, NPTS, OPT(8), XNAME, YNAME)
      ENDIF
      CLOSE(UNIT = 19)
      STOP
      END
C
C
C     ************************************************************
```

```
C
C   REPEAT
C
C   SUBROUTINE REPEAT IS THE SUBROUTINE THAT LOOPS THROUGH MARQUARDTS
C   ALGORITHM AND CHECK TO SEE IS THE SUM OF SQUARES HAS DECREASED FOR
C   THIS PARTICULAR LOOP.  IF IT HAS DECREASED THEN THE NEW PARAMETERS
C   REPLACE THE OLD ONES AND LAMBDA IS REDUCED BY A FACTOR OF 10.
C   IF THE SUM OF SQUARES IS LARGER
C   IF THE SUM OF THE SQUARES IS LARGER THEN THE OLD PARAMETERS ARE
C   RETAINED AND LAMBDA IS INCREASED BY A FACTOR OF 10.
C
      SUBROUTINE REPEAT (X,Y,NPTS,NTERMS,B,DELTAB, SIGMAB, LAMBDA,
     1                   YCALC, SUMSQR)
      DOUBLE PRECISION INVERSE(30,30), BETA(30), ALPHA(30,30)
      REAL X(1000,10), Y(1000), DERIV(30),B(30)
      REAL SIGMAB(30), YCALC(1000)
      REAL DELTAB(30)
      REAL LAMBDA, TEMPB(30)
C
C
C   SUMSQR INITIAL
C
      DO 60 I = 1,NPTS
60    YCALC(I) = FUNCTN(X,I,B)
      SUMSQ1 = FUNKOFF(Y,NPTS,YCALC)
C
C   EVALUATE MATRICTES
C
      DO 34 J=1,NTERMS
      BETA(J) = 0.D0
      DO 34 K = 1,J
34    ALPHA(J,K) = 0.D0
C
41    DO 50 I = 1,NPTS
      CALL FPRIME (X, I, B, DELTAB, NTERMS, DERIV)
      DO 46 J = 1,NTERMS
      BETA(J) = BETA(J) + 1.D0*(Y(I) - YCALC(I))*DERIV(J)
      DO  46 K = 1,J
46    ALPHA(J,K) = ALPHA(J,K) + 1.D0*DERIV(J)*DERIV(K)
50    CONTINUE
51    DO 53 J = 1,NTERMS
52    DO 53 K = 1,J
53    ALPHA(K,J)  = ALPHA(J,K)
C
C   INVERT
C
      WRITE (19, 55)
55    FORMAT('0',T16,'INITIAL MATRIX/VECTOR VALUES')
      WRITE (19, 56)
56    FORMAT('0')
      DO 57 J = 1,NTERMS
57    WRITE (19, 58) J, BETA(J), J, DERIV(J)
58    FORMAT(' ', 'BETA(',I2,')= ',G20.12,6X,'DERIV(',I2,')= ',G20.12)
      WRITE(19, 56)
      DO 59 J = 1,NTERMS
59    WRITE(19, 61) ((' ', I, J, ALPHA(I,J)), I = 1,J)
61    FORMAT(' ',10 (A1, 'ALPHA(', I2, ',',I2,')= ', G20.12,3X))
C
71    DO 74 J = 1,NTERMS
      DO 73 K = 1,NTERMS
73    INVERSE(J,K) = ALPHA(J,K)
```

```
74      INVERSE(J,J) = (1.D0 + 1.D0*LAMBDA)*ALPHA(J,J)
80      CALL IMATRIX( INVERSE, NTERMS, DET)
C
C
        WRITE(19, 21)
21      FORMAT('0', T16, 'INVERSE ALPHA MATRIX')
        WRITE(19, 56)
        DO 22 J = 1,NTERMS
22      WRITE(19, 61) ((' ', I, J, INVERSE(I,J)), I = 1,J)
        WRITE(19, 23)
23      FORMAT('0', T5, 'PARAMETER CHANGES')
        DO 240 I = 1,NTERMS
240     TEMPB(I) = 0.D0
        DO 24 I = 1,NTERMS
        DO 24 J = 1,NTERMS
24      TEMPB(I) = BETA(J)*INVERSE(I,J) + TEMPB(I)
        WRITE(19,30)
        DO 28 J = 1,NTERMS
28      WRITE(19,29) J, B(J), TEMPB(J),TEMPB(J) + B(J)
29      FORMAT(' ', ' ', I2, T10, G17.6, ' + ', G17.9, ' = ',G17.9)
30      FORMAT('0',T10,'ORIGINAL B',T30,'PLUS CHANGE',T50,'GIVES NEW B')
81      DO 84 J = 1,NTERMS
        TEMPB(J) = B(J)
        DO 84 K = 1,NTERMS
84      TEMPB(J) = TEMPB(J) + BETA(K) * INVERSE(J,K)
91      DO 92 I = 1,NPTS
92      YCALC(I) = FUNCTN(X,I,TEMPB)
93      SUMSQR = FUNKOFF(Y, NPTS, YCALC)
        SQ1R = SUMSQ1 - SUMSQR
        IF ( SQ1R .LE. 0.0) GOTO 95
        IF ( SQ1R .GT. 0.0) GOTO 101
95      LAMBDA = LAMBDA * 10.0
        GO TO 71
101     DO 103 J = 1,NTERMS
        B(J) = TEMPB(J)
103     SIGMAB(J) = 1.D0*SQRT(DABS(INVERSE(J,J)))
        LAMBDA = 0.10* LAMBDA
110     RETURN
        END
C
C
C
C  ******************************************************************
C
C  FPRIME
C
C  FPRIME IS THE SUBROUTINE THAT CALCULATES THE FUNCTION DERIVATIVES
C  BY DIFFERENCE METHODS.  DELTA B IS ADDED TO THE PARAMETERS AND THIS
C  USED WITH THE YCALC VALUES TO OBTAIN A VALUE FOR THE SLOPE WHICH
C  IS RETAINED AS A CLOSE APPROXIMAITON OF THE FUNCITON DERIVATIVE.
C
C
        SUBROUTINE FPRIME(X, I, B, DELTAB, NTERMS, DERIV)
        REAL X(1000,10),B(30),DELTAB(30), DERIV(30)
        DO 18 J = 1,NTERMS
        AJ = B(J)
        DELTA = DELTAB(J)
        B(J) = AJ + DELTA
        YFIT = FUNCTN(X,I,B)
        B(J) = AJ - DELTA
        DERIV(J) = (YFIT - FUNCTN(X,I,B))/(2*DELTA)
```

```
16        B(J) = AJ
          RETURN
          END
C
C
C
C    ****************************************************************
C
C    FUNKOFF
C
C    SUBROUTINE FUNKOFF CALCULATES THE SUM OF THE SQUARES OF THE FUNCTION
C    BY CALCULATING THE  VALUES OF THE INDEPENDANT VARIABLES AND SUBTRAACT
C    THESE FROM THE EXPERIMENTAL VALUES. THESE VALUES ARE THEM SQUARED
C    AND THEN SUMMED TOGETHER.
C
C
          FUNCTION FUNKOFF(Y, NPTS, YCALC)
          REAL Y(200), YCALC(200)
          FUNKOFF = 0.0
          DO 1 N = 1,NPTS
1         FUNKOFF = FUNKOFF + (Y(N) - YCALC(N))*(Y(N) - YCALC(N))
          RETURN
          END
C
C
C
C    ****************************************************************
C
C    IMATRIX
C
C    IMATRIX IS A MATRIX INVERSION SUVROUTTINE THAT INVERTS THE ALPHA
C    MATRIX TO BE USED IN THE SOLUTION OF THE  PARAMETER CORRECTION
C    VALUES.  TAKEN MOSTLY FROM BEVINGTON, DATA REDUCTION AND ERROR
C    ANALYSIS FOR THE PHYSICAL SCIENCES.
C
C
          SUBROUTINE IMATRIX(INVERSE,NORDER,DET)
          DOUBLE PRECISION INVERSE, MAXIMUM, SAVE
          DIMENSION INVERSE(30,30), IK(10),JK(10)
10        DET = 1.0
11        DO 100 K = 1,NORDER
C
C    FIND LARGEST ELEMENT OF MATRIX
C
          MAXIMUM = 0.D0
21        DO 30 I = K,NORDER
22        DO 30 J = K,NORDER
23        IF (DABS(MAXIMUM) - DABS(INVERSE(I,J)))24,24,30
24        MAXIMUM = INVERSE(I,J)
          IK(K) = I
          JK(K) = J
30        CONITNUE
C
C    INTERCHANGE ROWS AND COLUMNS IN ORDER TO INCREASE PRECISION
C
31        IF (MAXIMUM .EQ. 0.D0) CALL ERROR(-4)
          I = IK(K)
          IF (I-K) 21,51,43
43        DO 50 J=1,NORDER
          SAVE = INVERSE(K,J)
          INVERSE(K,J) = INVERSE(I,J)
```

```
50        INVERSE(I,J) = SAVE*(-1.D0)
51        J = JK(K)
          IF (J-K)  21, 61, 53
53        DO 60 I = 1,NORDER
          SAVE = INVERSE(I,K)
          INVERSE(I,K) = INVERSE(I,J)
60        INVERSE(I,J) = SAVE*(-1.D0)
C
C   ACCUMULATE ELEMENTS OF INVERSE MATRIX
C
61        DO 70 I = 1,NORDER
62        IF (I-K) 63, 70, 63
63        INVERSE(I,K) = (-1.D0)*INVERSE(I,K)/MAXIMUM
70        CONTINUE
71        DO 80 I = 1,NORDER
72        DO 80 J = 1,NORDER
73        IF (I-K) 74, 80, 74
74        IF (J-K) 75, 80, 75
75        INVERSE(I,J) = INVERSE(I,J) + INVERSE(I,K)*INVERSE(K,J)
80        CONTINUE
81        DO 90 J = 1,NORDER
          IF (J-K) 83,90,83
83        INVERSE(K,J) = INVERSE(K,J)/MAXIMUM
90        CONTINUE
          INVERSE(K,K) = 1.D0/MAXIMUM
100       DET = DET*MAXIMUM
C
C
C   RESTORE ORDERING OF MATRIX
C
101       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 = INVERSE(I,K)
          INVERSE(I,K) = (-1.D0)*INVERSE(I,J)
110       INVERSE(I,J) = SAVE
111       I = JK(K)
112       IF (I-K) 130, 130, 113
113       DO 120 J=1,NORDER
          SAVE = INVERSE(K,J)
          INVERSE(K,J) = (-1.D0)*INVERSE(I,J)
120       INVERSE(I,J) = SAVE
130       CONTINUE
140       RETURN
          END
C
C
C   ******************************************************************
C
C   FITTER
C
C
C   SUBROUTINE FITTER DOES THE ACTUAL LOOPING FOR THE PROGRAM.
C
          SUBROUTINE FITTER
          COMMON/NAME/BNAME
          COMMON/INSTUFF/ NIV, NTERMS, NPTS,NDEL,B,DELTAB,X,Y
          COMMON/RESULT/YCALC,SUMSQR,SIGMAB,ITER,LAMBDA
          COMMON/OPTIONS/CLAM,SLIM,SQUP,RLAM
```

```
              COMMON/OPT/OPT
              REAL YCALC(1000),SUMSQR, LAMBDA, LAM
              INTEGER OPT(30), SOOP, SLIM
              CHARACTER*4 TITLE(9), HEAD(9), DNAME(1000),OPTION(30)
              CHARACTER*4 BNAME(30,3), XNAME(10,3), YNAME(3)
              INTEGER NDELETE(30)
              REAL Y(1000), X(1000,10),B(30),DELTA B(30), BOLD(10)
              REAL BSAVE(10), SIGMAB(30)
              LAMBDA = 10.
              LOOPS = 30
              LIMIT = 5
              IF(OPT(5) .EQ. 1) LAMBDA = CLAM
              IF(OPT(6) .EQ. 1) LIMIT = SLIM
              IF(OPT(7) .EQ. 1) LOOPS = SOOP
              FLAG = 0
              ITER = 0
              DO 10 N=1,NTERMS
              BSAVE(N) = B(N)
    10        BOLD(N) = 0.0
              DFREE = NPTS - NTERMS - 1.0
    60        DO 61 N =1,NPTS
    61        YCALC(N) = FUNCTN(X,N,B)
              SOLD = FUNKOFF(Y,NPTS,YCALC)
              IF (OPT(1) .EQ. 1) WRITE(19,93) SOLD
    93        FORMAT('0', 'THE INITIAL SUM OF SQUARES IS ', G18.10)
              DO 11 I = 1,LOOPS
              ITER = ITER + 1
              CALL REPEAT(X,Y,NPTS,NTERMS[illegible]ELTAB,SIGMAB,LAMBDA,YCALC,SUMSQR)
              A1 = SOLD - SUMSQR
              XI1 = ALOG10(SOLD)
              I1 = INT(XI1)
              B1 = 10.0**I1
              SIGFIG = -1.0*(ALOG10(A1/B1))
              IF (OPT(1) .EQ. 0) GOTO 67
              WRITE(19,62) ITER
    62        FORMAT('0',3(/),' ', 'RESULTS OF ITERATION #' ,I3)
              WRITE(19,63) SUMSQR, SQRT(SUMSQR/DFREE)
    63        FORMAT(' ','THE SUM OF SQUARES IS ', G18.10,6X,
         1    'THE AVERAGE RESIDUAL IS ', G18.10)
              WRITE(19,64)
    64        FORMAT('0',T6,'PARAMETER',14X,'VALUE',17X,'STANDARD ERROR')
              WRITE(19,70)
    70        FORMAT(' ')
              DO 65 N = 1,NTERMS
              WRITE(19,66) N,(BNAME(N,L),L=1,3),B(N),SIGMAB(N)
    66        FORMAT(' ',  I3, T6, 3(A4), 6X, G18.10, 6X, G18.10)
    65        CONTINUE
              WRITE(19,70)
              CALL LINE(1)
              WRITE(19,70)
    67        SOLD = SUMSQR
              IF (SIGFIG .GE. LIMIT) WRITE(19,100)
   100        FORMAT(' ', 'NORMAL CONVERGENCE ')
              IF (SIGFIG .GE. LIMIT) GOTO 2
    11        CONTINUE
              IF(FLAG .EQ. 1) CALL ERROR(-5)
              LAM = 100000.
              CALL LINE(1)
              IF (OPT(9) .EQ. 1) LAM = RLAM
              WRITE(19,68) ITER, LAM
    68        FORMAT(' ', I3, ' ITERATIONS AND NO CONVERGENCE.    ',
```

```
     1  'INITAL PARAMETERS RESTORED ND LAMBDA SET TO ',F10.3 )
        CALL LINE(1)
        LAMBDA = LAM
        DO 69 N = 1,NIFRMS
69      B(N) = BSAVE(N)
        FLAG = 1
        GOTO 60
2       RETURN
        END
C
C
C   ********************************************************************
C
C   HEADING
C
C   SUBROUTINE PRINTS INFORMATION ON RUN
C
        SUBROUTINE HEADING (TITLE, HEAD)
        INTEGER TITLE(9), HEAD(9)
        WRITE(19,6) (HEAD(I), I=1,9)
        WRITE(19,8) (TITLE(J), I=1,9)
6       FORMAT ('0', T38, 9A4)
8       FORMAT ('0', T40, 94A)
        RETURN
        END
C
C
C   ********************************************************************
C
C   LINE
C
C   SUBROUTINE LINE SIMPLY PRONTS OUT THE REQUESTED NUMBER OF HEAVILY
C   OVERPRINTED LINES USED IN BOTH THE HEADING SUBROUTINE AND THE ERROR
C   SUROUTINE.
C
        SUBROUTINE LINE(N)
        DO 1 M = 1,N
        WRITE(19,2)
        WRITE(19,3)
        WRITE(19,4)
        WRITE(19,5)
1       WRITE(19,6)
2       FORMAT(' ',  55('MM'))
3       FORMAT('+', 55('WW'))
4       FORMAT('+', 55('XX'))
5       FORMAT('+', 55('(('))
6       FORMAT('+', 55('))'))
        RETURN
        END
C
C
C   ********************************************************************
C
C   INOUT
C
C   SUBROUTINE INOUT PRINTS OUT ALL OF THE INPUT DATA EXCEPT THE ACTUAL
C   DATA SET ITSELF THAT WOULD BE OF INTEREST AND THAT COULD CHANGE
C   BETWEEN RUNS.
C

        SUBROUTINE INOUT
```

```
      COMMON/NAME/BNAME
      COMMON/INSTUFF/ NIV, NTERMS, NPTS, NDEL,B,DELTAB,X,Y
      REAL X(1000,10),Y(1000),DELTAB(30),B(30)
      INTEGER NIV,NTERMS,NPTS,NDEL
      CHARACTER*4 BNAME(30,3)
      WRITE(19,1)
1     FORMAT ('0', 48('*'), '   DATA INPUT   ',48('*'))
      WRITE(19,7)
      WRITE(19,2) NIV,NTERMS
2     FORMAT ('  ', 12X, I2, '  INDEPENDENT VARIABLES',37X,I2,
     1   '         EQUATION PARAMETERS')
      WRITE(19,4)
4     FORMAT('0', 42('*'), '  INITIAL DATA PARAMETERS  ',41('*'))
      WRITE(19,7)
      WRITE(19,5)
5     FORMAT(' ',27X,'NUMBER',8X,'B NAME',8X,'B VALUE',9X,'DELTA B')
      WRITE(19,6)
6     FORMAT(' ',27X,6('-'),8X,7('-'),7X,7('-'),9X,7('-'))
      WRITE(19,7)
      L = 0
      DO 8 M = 1,NTERMS
      L = L + 1
      IF(L .LT. 6 .AND. M-2 .LT. NTERMS) GOTO 8
      L = 0
      WRITE(19,7)
7     FORMAT(' ')
8     WRITE(19,9) M, (BNAME(M,J),J=1,3), B(M), DELTAB(M)
9     FORMAT(' ',28X,I4,6X,3(A4),4X,G12.5,4X,G12.5)
      RETURN
      END
C
C
C
C  ****************************************************************
C
C  RESID
C
C  SUBROUTINE RESID PRINTS OUT A LISTING OG THE RESIDUALS IN BOTH
C  INITIAL -READ IN- ORDER AND IN ORDER OF THE ASCENDING RESIDUAL
C  VALUES IN ORDER FOR PROBLEMS WITH THE DATA SET TO COME TO THE
C  ATTENTION OF THE USER MORE RAPIDLY.
C
C

C
      SUBROUTINE RESID(Y, YCALC, DNAME, TITLE, NPTS)
      REAL Y(1000),YCALC(1000),RES(1000)
      INTEGER ORDER(1000),TITLE(9)
      CHARACTER*4 DNAME(1000)
      WRITE(19,3)
      WRITE(19,4)  (TITLE(J),J=1,9)
      WRITE(19,6)
      WRITE(19,5)
      WRITE(19,6)
      WRITE(19,11)
      WRITE(19,6)
      WRITE(19,7)
      WRITE(19,8)
      WRITE(19,6)
      WRITE(19,10)
      DO 1 N =1,NPTS
1     RES(N) = Y(N) - YCALC(N)
      CALL SORT(RES, ORDER, NPTS)
```

```
      I = 0
      DO 2 N = 1,NPTS
      I = I+1
      M = ORDER(N)
      WRITE(19,9) N,DNAME(N),Y(N),YCALC(N),RES(N),M,DNAME(M),Y(M),
     1  YCALC(M),RES(M)
      IF (I .LT. 6) GOTO 2
      I = 0
      WRITE(19,10)
2     CONTINUE
      WRITE(19,10)
      WRITE(19,6)
3     FORMAT ('1',T39,'NON-LINEAR LEAST SQUARES DATA FIT V3.0')
4     FORMAT ('0', T40,9A4)
5     FORMAT (' ', '*', 55X, 'RESIDUALS', 55X, '*')
6     FORMAT(' ', 121('*'))
7     FORMAT(' ',2('* ',8X,'DATA',7X,'DATA',8X,'CALCULATED',6X,
     1 'RESIDUAL',4X),'*')
8     FORMAT (' ', 2('* ORDER NAME',7X,'VALUE',9X,'VALUE',21X),
     1 '*')
9     FORMAT(' ',2('*  ',I4,2X,A8,2X,3(G13.6,2X)),'*')
10    FORMAT(' ','*',59(' '),'*',59(' '),'*')
11    FORMAT(' ','*',21X,'ORDERED BY INPUT',22X,'*',
     1  19X,'ORDERED BY RESIDUALS',20X,'*')
      RETURN
      END
C
C
C  ****************************************************************
C
C  SORT
C
C  SUBROUTINE SORT IS A SIMPLE BUBBLE SORT SUBROUTINE THAT RETURNS
C  THE ASCENDING ORDER IN THE ARRAY LABLED ORDER.
C
      SUBROUTINE SORT(X,ORDER,NUMBER)
      REAL X(1000),Y(1000),Y1
      INTEGER ORDER(1000),NUMBER,N1
C
C  INITIALIZE ORDER ARRAY AND PLACE ALL X VALUES INTO THE CORRESPONDING
C
      DO 1 N = 1,NUMBER
      Y(N) = X(N)
1     ORDER(N) = N
C
C  SET FLAG AND LOOP PARAMETERS
C
      M = NUMBER - 1
      FLAG = 0
C
C  RUN THROUGH LOOP UNTIL A LOOP OCCURS THAT NO ACTION WAS TAKEN OR ELSE
C  THE END OF THE LOOP HAS BEEN REACHED.
C
2     DO 3 N = 1,M
      IF (Y(N+1) .GE. Y(N)) GOTO  3
      N1 = ORDER(N)
```

```
      ORDER(N) = ORDER(N+1)
      ORDER(N+1) = N1
      Y1 = Y(N)
      Y(N) = Y(N+1)
      Y(N+1) = Y1
      FLAG = 1
3     CONTINUE
      M = M - 1
      IF (FLAG .EQ. 0 .OR.  M .EQ. 0) RETURN
      FLAG = 0
      GOTO 2
      END
C
C
C  *************************************************************
C
C  OUTPUT
C
C  SUBROUTINE OUTPUT PRINTS OUT THE RESULTS OF THE MARQUARDTS METHOD
C
C
      SUBROUTINE OUTPUT
      COMMON/NAME/BNAME
      COMMON/INSTUFF/ NIV,NTERMS,NPTS,NDEL,B,DELTAB,X,Y
      COMMON/RESULT/YCALC,SUMSQR,SIGMAB,ITER,LAMBDA
      REAL YCALC(1000),SUMSQR,SIGMAB(30),LAMBDA
      REAL Y(1000), X(1000,10), B(30), DELTAB(30)
      INTEGER  ITER
      CHARACTER*4 BNAME(30,3)
      WRITE(19,1)
      WRITE(19,2)
      WRITE(19,3)
      DO 8 I = 1,NTERMS
8     WRITE(19,9) I, (BNAME(I,J), J=1,3), B(I), SIGMAB(I)
      WRITE(19,10)  ITER, SUMSQR, SQRT(SUMSQR/NPTS)
      WRITE(19,2)
1     FORMAT('1')
2     FORMAT('0',45('-'),'  PROGRAM RESULTS  ', 46('-'))
3     FORMAT('0',40('*'),'  FITTED PARAMETER INFORMATION  ',39('*'))
9     FORMAT('0', 20X, I3, 6X, 3A4, 4X, G12.5, 2X, G12.5)
10    FORMAT('0', ' A TOTAL OF ',
     1  I2, ' ITERATIONS.  THE RESIDUAL SUM OF SQUARES IS ',
     1  G18.10, ' THE STANDARD DEVIATION IS ',G18.10)
      RETURN
      END
C
C
C  ******************************************************************
C
C  PLOT
C
C  SUBROUTINE PLOT IS USED TO GENERATE PLOTS OF FUNCTIONS AND RESIDUAL
C  THAT ARE CALLED FROM ELSEWHERE IN THE PROGRAM.
C
      SUBROUTINE PLOT(X,Y,NUMBER,OPT)
      REAL X(1000),Y(1000)
      INTEGER OPT, NUMBER
      CHARACTER*1 C(11), A(111)
      DATA  C/' ','1','2','3','4','5','6','7','8','9','X'/
      XMIN = X(1)
      XMAX = X(1)
```

```
      YMIN = Y(1)
      YMAX = Y(1)
      DO 1 N = 1,NUMBER
      IF (X(N) .GT. XMAX) XMAX =  X(N)
      IF (YMAX .LT. Y(N)) YMAX =  Y(N)
      IF (YMIN .GT. Y(N)) YMIN =  Y(N)
1     IF (X(N) .LT. XMIN) XMIN =  X(N)
      DY = (YMAX - YMIN)/20.0
      DX = (XMAX - XMIN)/50.0
      XMAX = XMAX + DX
      XMIN = XMIN - DX
      YMIN = YMIN - DY
      YMAX = YMAX + DY
      DX = (XMAX - XMIN)/111.0
      DY = (YMAX - YMIN)/51.0
      WRITE(19,42)
      COUNT = -1
      DO 11 N1 = 1,51
      DO 40 N = 1,111
40    A(N) = ' '
      B = 'I'
      COUNT = COUNT + 1
      IF (COUNT .EQ. 5) B = '+'
      IF (COUNT .EQ. 5) COUNT = 0
      IF (N1 .EQ. 1) B = '+'
      YVAL1 = YMAX - DY*(N1-2)
      YVAL2 = YMAX - DY*(N1-1)
      DO 12 M1 = 1,NUMBER
      IF (Y(M1) .LT. YVAL2 .OR. Y(M1) .GE. YVAL1) GOTO 12
      DO 13 N2 = 1,111
      XVAL1 = XMIN + DX*(N2-1)
      XVAL2 = XMIN + DX*(N2)
      IF (X(M1) .LT. XVAL1 .OR. X(M1) .GE. XVAL2) GOTO 13
      DO 61 L = 1,10
      IF (A(N2) .NE. C(L)) GOTO 61
      A(N2) = C(L+1)
      GOTO 12
61    CONTINUE
      GOTO 12
13    CONTINUE
12    CONTINUE
      WRITE(19,20) YVAL2, B, (A(M), M=1,111),B
20    FORMAT (' ', G14.6,T17,113(A1))
      IF (OPT .NE. 1) GOTO 11
      SIGN = YVAL1*YVAL2
      IF (SIGN .GT. 0) GOTO 11
21    FORMAT (' ', T17, 113('-'))
11    CONTINUE
      WRITE(19,42)
42    FORMAT (' ', T16, 11('+---------'), '+')
      WRITE(19,43) ((XMIN+(J-1)*111*DX/11), J=1,11,2)
      WRITE(19,44) ((XMIN+(J-1)*111*DX/11), J=1,12,2)
43    FORMAT (' ', T12, 6(G15.8,5X))
44    FORMAT (' ', T12, 6(G15.8,5X))
      RETURN
      END
C
C
C
C   ***********************************************************
C
C   RESTOP
```

```
C
C  SUBROUTINE RESIDP PRINTS OUT PLOTS OF THE RESIDUALS IN ALMOST
C  EVERY CONCEIVABLE ORDER IN ORDER FOR THE USER TO DISCOVER ANY
C  PATTERNS IN THE RESIDUALS.  THIS IS PROBABLY THE MOST USEFUL
C  PROBLEM DETECTING OPITON IN THE WHOLE PROGRAM.
C
       SUBROUTINE RESIDP(Y,X,NIV,YCALC,NPTS,SMOD,XNAME,YNAME)
       REAL XX(1000), RES(1000), Y(1000), YCALC(1000), X(1000,10)
       REAL XP(1000)
       INTEGER ORDER(1000),SMOD
       CHARACTER*4 XNAME(10,3),YNAME(3)
       DO 1 N = 1,NPTS
       XX(N) = N
       ORDER(N) = N
       RES(N) = Y(N) - YCALC(N)
1      CONTINUE
       IF(NIV .EQ. 1) THEN
       WRITE(19,35) (YNAME(M), M=1,3), (XNAME(1,I), I=1,3)
35     FORMAT('1',2(/),T41,3(A4),' AGAINST ',3(A4))
       DO 36 K=1,NPTS
36     XP(K) = X(K,1)
       CALL PLOT(XP,Y,NPTS,1)
       ENDIF
       WRITE(19,10)
10     FORMAT('1',2(/),T50,' RESIDUALS PLOT')
       WRITE(19,11)
11     FORMAT(' ', T42, 'RESIDUALS AGAINST INPUT ORDER')
       CALL PLOT(XX, RES, NPTS, 1)
       WRITE(19,10)
       WRITE(19,31) (YNAME(M),M=1,3)
31     FORMAT(' ',T42,'RESIDUALS AGAINST ',3A4)
       CALL PLOT (Y,RES,NPTS,1)
       DO 32 N = 1,NIV
       WRITE(19,10)
       WRITE(19,33) (XNAME(N,I),I=1,3)
33     FORMAT(' ', T42, 'RESIDUALS AGAINST ', 3(A4))
       DO 34 M = 1,NPTS
34     XX(M) = X(M,N)
       CALL PLOT(XX, RES, NPTS, 1)
32     CONTINUE
       CALL SORT (RES, ORDER, NPTS)
       DO 2 N = 1,NPTS
2      XX(N) = RES(ORDER(N))
       DO 801 N = 1,NPTS
       RES(N) = XX(N)
801    XX(N) =  N
       WRITE(19,10)
       WRITE(19,22)
22     FORMAT(' ', T42, 'RESIDUALS PLOTTED IN ORDER')
       CALL PLOT (XX, RES, NPTS, 1)
       RETURN
       END
C
C  **************************************************************
C
C  ERROR
C
C  SUBROUTINE ERROR IS USED TO GENERATE SELFEXPLAANITORY OUTPUT TO
C  TELL THE USER WHAT WENT WRONG IN THE PROGRAM.
C
```

```
      SUBROUTINE ERROR(NE)
      CHARACTER*15 TELL1(10), TELL2(10)
      DATA TELL1(1)/ 'DEGREES OF'/
      DATA TELL2(1)/ ' FREEDOM  '/
      DATA TELL1(2)/ 'PARAMETER '/
      DATA TELL2(2)/ 'INPUT READ '/
      DATA TELL1(3)/ 'DATA POINT'/
      DATA TELL2(3)/ ' READ IN '/
      DATA TELL1(4)/ 'A MATRIX S'/
      DATA TELL2(4)/ 'INGULARITY'/
      DATA TELL1(5)/ 'EXCEED MAX'/
      DATA TELL2(5)/ ' ITER #   '/
      DATA TELL1(6)/ 'ILLEGAL OP'/
      DATA TELL2(6)/ 'TION VALUE'/
      DO 4 M = 1,4
      WRITE(19,1)
1     FORMAT(' ')
      CALL LINE(1)
      IF (M .EQ. 3) WRITE(19,2) IABS(NE)
      IF (M .EQ. 3 .AND. NE .LT. 0) WRITE(19,3)
2     FORMAT ('0', 'ERROR NUMBER ', I2, ' DETECTED BY DATAFIT')
3     FORMAT ('0', 'THIS IS A FATAL ERROR')
4     CONTINUE
      WRITE(19,5)
5     FORMAT ('0', ' AN ERROR HAS OCCURED IN YOUR DATAFIT PROGRAM', /,
     1 '0IF THE ERROR IS NOT FATAL AN ATTEMPT WILL BE MADE TO RECOVER')
      WRITE(19,6) TELL1(IABS(NE)), TELL2(IABS(NE))
6     FORMAT ('0', 'THE ERROR THAT OCCURED WAS IN ', 2(A10))
      DO 7 M = 1,3
      WRITE(19,1)
7     CALL LINE(1)
      IF (NE .LT. 0) STOP
      RETURN
      END
C
C
C ******************************************************************
C
C FUNCTN
C
C THIS IS WHERE THE USER SUPPLIED SUBROUTINE IS TO GO. IT MUST HAVE
C THE EXACT SAME FIRST TWO LINES AS IN THE EXAMPLE  SUBROUTINE LISTED
C BELOW AND LIST THE RESULT INTO THE VARIABLE FUNCTN.
C
      FUNCTION FUNCTN (X, I, B)
      REAL X(1000,10), B(30)
      F = X(I,1)
C     T = X(I,2)
C     FUNCTN = B(1)*ALPHA**2 - B(2)*ALPHA + B(3)
C     FUNCTN =(B(1) + B(2)*T)*EXP(-(B(3)/(T*A))
C     FUNCTN = B(1) + B(2)*(1-F)/F
      FUNCTION = B(1) + B(2)*F
      RETURN
      END
C
C
C ***************************************************************
C
C
      SUBROUTINE SIGPAR (SUMSQR,X,NPTS)
      REAL X(1000,10)
```

```
      SUMX2 = 0
      SUMX = 0
      DO 1 I = 1,NPTS
      SUMX2 = SUMX2 + X(I,1)*X(I,1)
1     SUMX = SUMX + X(I,1)
      F = NPTS*SUMX2 - SUMX**2
      SIGA = SUMX2*SUMSQR/(NPTS-2)
      SIGA = SQRT(SIGA)
      SIGB = SUMSQR*NPTS/(F*(NPTS-2))
      SIGB = SQRT(SIGB)
      WRITE(19,200) SIGA, SIGB
200   FORMAT(' ',' UNCERTIANTY IN PARAMETER A = ',1PE11.4,2(/),
     1 ' UNCERTIANTY IN PARAMETER B = ',1PE11.4)
      RETURN
      END
```

REFERENCES

[1] D'Amato, R. J., Ph.D. Thesis, University of Illinois, Urbana, Illinois (1971).

[2] Hay, A. J., Ph.D. Thesis, University of Illinois, Urbana, Illinois (1966).

[3] Trafton, W. E., Ph.D. Thesis, University of Illinois, Urbana, Illinois (1973).

[4] Wei, G., Ph.D. Thesis, University of Illinois, Urbana, Illinois (1974).

[5] Callear, A. B., Tyerman, W. J. R., *Trans. Faraday Soc.*, *61*, 2395 (1965).

[6] Lebreton, J., Bosser, G., Marsigny, L., *J. Phys. B: Atom. Molec. Phys.*, *6*, L226 (1973).

[7] Ashford, M. N. R., Quinton, A. M., Simons, J. P., *J. Chem. Soc. Faraday Trans.* II, 76 (8), 903 (1981).

[8] Marquart, J. R., Belford, R. L., Fraenkel, H. A., *Intern. J. Chem. Kinet.*, *9*, 671 (1977).

[9] Belford, R. L., Marquart, J. R., Fraenkel, H. A., *Intern. J. Chem. Kinet.*, *11*, 1109 (1979).

[10] King, G. W., Srikameswaran, K., *J. Molec. Spectrosc.*, *31*, 269 (1969).

[11] Graziano, L. C., Ph.D. Thesis, University of Illinois, Urbana, Illinois (1982).