

INTERNAL REPORT SDL: 92-01

June 1992

Absorption Spectroscopy at SDL

K. G. Casey

**Shock Dynamics Laboratory
Department of Physics
Washington State University
Pullman, WA 99164-2814**

TABLE OF CONTENTS

LIST OF TABLES	ii
LIST OF FIGURES	iii
1.0 INTRODUCTION	1
2.0 EXPERIMENTAL EQUIPMENT	1
2.1 EXPERIMENTAL PROCEDURES	1
3.0 SYSTEM EVALUATION.....	2
3.1 SPECTRAL RESPONSE	3
3.2 SYSTEM TESTING.....	4
4.0 ERROR ANALYSIS.....	5
5.0 REFERENCES.....	7
APPENDIX A: Lens Collection System.....	35
APPENDIX B: Reflection Losses.....	39
APPENDIX C: Data Analysis	44
C.1 Introduction.....	44
C.2 Beer's Law.....	44
C.3 Computer Programs	48
C.3.1 <i>ABSFIT</i> Program	48
C.3.2 <i>MAK-ABS-DATA</i> Program	90
APPENDIX D: Error Propagation	100

LIST OF TABLES

1	Components.....	9
2	Spectral Efficiencies.....	10
B.1	Refraction Indices	41
C.1	<i>ABSFIT</i> Routines	97

LIST OF FIGURES

1	Experimental Configuration.....	11
2	Spectral Distribution of Xenon Flashtube.....	12
3	Reflectance of UV Enhanced Mirrors.....	13
4	Transmittance of Sapphire	14
5	Transmittance of UV Grade Fused Silica	15
6	Attenuation of Optical Fibers	16
7	Reflectance of Aluminum Magnesium Fluoride Mirrors	17
8	Diffraction Grating Efficiency	18
9	Relative Sensitivity of Streak Camera	19
10	Spectral Response of Vidicon Detector.....	20
11	Detected Mercury Spectrum - Old Lamp	21
12	Detected Mercury Spectrum - K. Hipps' Lamp	22
13	Calibrated Mercury Spectrum - K. Hipps' Lamp	23
14	Xenon Flashtube Spectral Response - 300nm Grating	24
15	Xenon Flashtube Spectral Response - 500nm Grating.....	25
16	Xenon Flashtube Spectral Response - Shot #92-006, Hexane Filled Cell.....	26
17	Xenon Flashtube Spectral Response - Carbon Disulfide-Filled Cell.....	27
18	Xenon Flashtube Spectral Response - Shot #92-006, Nitromethane Filled Cell	28
19	Absorption Spectrum of Hexane	29
20	Absorption Spectrum of Carbon Disulfide.....	30
21	Absorption Spectrum of Nitromethane	31
22	Published Absorption Spectrum of Carbon Disulfide	32
23	Published Absorption Spectrum of Nitromethane.....	33
24	Absorption Spectrum of Nitromethane with Error Bars.....	34
A.1	Lens Configuration	37
A.2	Focusing on the Fiber End.....	38
B.1	Reflection Losses	42
B.2	Corrected Hexane Spectra.....	43
C.1	Relative Error versus Transmittancy.....	99

1.0 INTRODUCTION

References 1, 2, and 3 describe the absorption and reflection spectroscopy techniques in use at SDL. This internal report is a comprehensive description of these procedures. The following sections describe the experimental equipment, procedures used, system evaluation and capabilities, and error analysis. The appendices, besides containing the listings of the analysis programs, have specific details relevant to the absorption experiment.

2.0 EXPERIMENTAL EQUIPMENT

Figure 1 is a schematic view of the current experimental configuration used for an absorption measurement in a liquid. Details regarding each of the experimental components shown in Figure 1 are listed in Table 1. These details include information about each component, its manufacturer, model number, and any other important characteristics. These components are: (1) xenon flashlamp⁴, (2) off-axis, parabolic mirror⁵, (3) three flat turning mirrors⁶, (4) aperture⁷, (5) sapphire impactor⁸, (6) sample⁸, (7) two UV-coated fused silica lenses⁹, (8) optical fiber(s)¹⁰, (9) spectrometer¹¹, (10) streak camera¹², (11) vidicon detector and controller¹³, (12) optical multichannel analyzer (OMA)¹³, and (13) computer (not shown)¹⁴.

2.1 EXPERIMENTAL PROCEDURES

The flashlamp emits a light pulse of approximately 10 microseconds duration which is turned and collimated by the off-axis parabolic mirror. The light passes through a quartz window (not shown in Figure 1) into the sample area of the gun where the first flat mirror bends it into the gun barrel. The other two flat mirrors, attached to the projectile, reflect the light through the aperture, impactor, and sample. If absorption occurs, some of the UV-visible photons will be subtracted from the pulse. The light is collected and coupled into the optical fiber. Details regarding choice of lenses, determination of spot sizes, etc. is given in Appendix A.

The end of the fiber is placed in the focal plane of the spectrometer for maximum collection of photons. The spectrometer spectrally disperses the light and the streak camera temporally disperses the signal. The vidicon detector records the optical output which is then displayed on the OMA. The computer stores and analyzes the data.

3.0 SYSTEM EVALUATION

The efficiency of the system for a spectral range of 300-500nm is now considered. The manufacturer's spectral output⁴ of the xenon flashlamp is shown in Figure 2. The reflectance data for the UV-enhanced mirrors⁶ is shown in Figure 3 while the transmission data for sapphire⁹ is shown in Figure 4. The transmission data for the lenses⁹ is shown in Figure 5. The spectral attenuation of the fiber¹⁰ is shown in Figure 6. The reflectance of the spectrometer's mirrors⁹ is in Figure 7 while the grating efficiency¹⁵ is shown in Figure 8. The photocathode response of the streak camera¹² is shown in Figure 9. The calculation considers only optical components after the aperture and it is assumed the sample is an empty (air filled) liquid cell.

There are usually three causes of light attenuation as the photons pass through different components: two surface reflection losses and absorption by the component. The reflection losses can be calculated if the wavelength dependency of the refraction index is known. From either transmittance or absorption curves, the attenuation is determined. The losses are:

- (a) Sapphire impactor, front and back windows of the liquid cell: There are two surface reflection losses and the absorption of the sapphire. Approximately 8% is lost at each surface(reflection) and < 1.5% is absorbed (for a 3.175mm thick sample).
- (b) Fused silica lenses (two): Approximately 4% is lost at each surface and <0.5% is absorbed.
- (c) Optical fiber (the first one approx. 3' long and the second one approx. 15' long). Approximately 4% is lost at each end. The shorter fiber loses approximately 0.003-2.5% while the longer fiber loses 1.4-12% of the light due to absorption.
- (d) The three mirrors in the spectrometer: The loss of light is due to single reflections and is approximately 12% for each mirror.
- (e) The diffraction grating: The efficiency of the grating is dependent upon two factors: the reflectance of the aluminum and the efficiency of the blaze angle. The loss of light ranges from 34 to 61% and is wavelength dependent.
- (f) The streak camera: The efficiency of the streak camera photocathode is wavelength dependent and the loss of light ranges from 48 to 20%. Also, there are reflection/absorption losses due to the fused silica faceplate. The losses are approximately 13-15%.

Table 2 lists these efficiencies at three different wavelengths (300, 400, 500nm). The complete efficiency of the system at 300 nm is 5.4%, at 400 nm is 12%, and at 500 nm is 8.5%. Figure 10 shows the spectral response of the 1252B vidicon detector¹³. The anode

of the streak camera is a blue phosphorescent screen. Therefore, even though the original signal contains photons of many different wavelengths, only blue photons are emitted by this anode and recorded by the vidicon detector. All wavelengths of the original flash of light are seen with the same efficiency by the vidicon. For this reason, the vidicon efficiency is not included in Table 2. The figure is shown only for completeness of information.

3.1 SPECTRAL RESPONSE

A mercury (Hg) lamp is used for wavelength calibration of the system. When the lamp is (1) attached directly to the spectrometer via an optical fiber or (2) placed at the spectrometer's entrance slits, spectral lines below @ 310nm were not seen. Though the efficiency of the system at 300nm is low (and lower still for wavelengths <300nm), it should be possible to detect light with wavelength less than 300nm. On the possibility that the mercury lamp use was 'bad', i.e., lost pressure, impurities in gas, etc., a calibrated mercury lamp was borrowed from Dr. K. Hipps, Department of Chemistry, WSU¹⁶. Using this lamp, the system detected light down to 254nm. Figure 11 shows the response of the system to the original mercury lamp while Figure 12 shows the response for Dr. Hipps' lamp. Figure 13 is the calibrated, spectral output of Dr. Hipps' lamp. These figures indicate that the system is capable of 'seeing' from 250nm to 620nm. Its possible and very probable that the system can detect even longer wavelengths. To do so, a grating blazed at longer wavelengths must be used as well as a light source capable of emitting longer wavelength photons.

Comparing the spectral range shown in Figure 2 with Figure 12 implies that information below 300nm is obtainable. The spectral output of the xenon flashtube is shown in Figure 14. Immediately obvious is the peak intensity occurs @ 400nm (not 300-350nm) and the long wavelength tail has a sharper than expected intensity drop. Answering these questions involved many conversations with Xenon Corporation¹⁷.

This model of flashlamp has three configurations all of which utilize the same flashtube but have different size discharge capacitors. The sizes and corresponding energy storage are 2 microfarads (100 J), 0.5 microfarads (25 J), and 0.2 microfarads (10 J). The current flashlamp used is the 25 J capacitor model. The spectral curve (Figure 2) by Xenon Corporation is for the 100 J model. The 100 J capacitor delivers more energy causing the gas to be hotter and thereby blue-shifting the spectrum. (According to Xenon Corporation, the gas must be > 10000K to get the spectrum in Figure 2 while the 25 J model only heats

the gas to @ 6000-7000K). Since the gas is cooler, the spectral output is red-shifted. Xenon's engineers think Figure 14 is reasonable for the 25 J model.

An accurate spectral distribution curve for the 25J model flashlamp is not available. (On several occasions such a curve was requested from the Xenon Corporation. Their response was 'just shift the 100J curve by about 100nm'. Doing this, it appears the decrease in the longer wavelength region is still too abrupt. Unfortunately, this technique, while in general gives an appropriate, approximate curve, is not accurate enough to be trustworthy.)

Perhaps the best and simplest explanation involves the choice of grating. Replacing the current grating with one blazed for a longer wavelength (i.e., 500nm) shows a much slower tail-off in the visible wavelength region (see Figure 15). This recorded spectrum is similar to Figure 2 in the longer wavelength region. Note however that the intensity peak is now closer to 5000 angstroms. Obviously, the grating used plays a major role in the flashlamp's spectral output detected by the system.

3.2 SYSTEM TESTING

Solutions of known absorbance have been tested as well as artificial data. (The data was generated by the Fortran program *MAK-ABS-DATA* located in */users/kelly/bin*. To save space, the results are not shown here even though they confirmed the validity of the analysis programs. The *MAK-ABS-DATA* listing is shown in Appendix C). Solutions tested were hexane, carbon disulfide, and nitromethane. The transmission spectrum of these solutions are shown in Figures 16, 17, and 18. For comparison, the transmission of an empty (air filled) cell is also shown.

Hexane was used due to its small absorbance² at wavelengths > 300nm. Its transmission curve, while similar to the empty cell spectrum, is not exactly the same (see Figure 16). Two reasons contribute to this discrepancy: pulse to-pulse fluctuation of the flash lamp and reflection losses at the sapphire-hexane interface.

The pulse-to-pulse fluctuation can be as great as 8-10% but is minimized by consistent firing of the flashlamp. By experimenting, it is best to fire the lamp at intervals of two minutes. The reflection loss is dependent upon the refractive indices of the solution and the sapphire. The closer the liquid's index is to that of sapphire, the less reflection loss. By knowing the refractive index of the liquid, these reflection losses may be corrected. Details of correcting these reflection losses are given in Appendix B.

Carbon disulfide was used due to the amount of prior work done on it at SDL. The absorption spectrum is well-known and has been measured using the current system².

Nitromethane is used because it is the material most likely to be studied for the next several years. Also, its absorption spectrum is known¹⁸.

Absorption spectra were calculated using the computer programs described in Appendix C. These spectra are shown in Figures 19 (hexane), 20 (carbon disulfide), 21 (nitromethane). The 'peaks' in the carbon disulfide and nitromethane plots are misleading. The system is capable of measuring absorbances of approximately < 1.0 (assuming a reference transmission of 2800 counts and a sample transmission of 280 counts). For many pure substances, the molar absorptivity is on the order of 10000. To measure an absorption spectrum of nitromethane and detect the peak absorbance, the cell width or solution concentration must be decreased. It is easier to decrease the concentration. If the concentration is decreased to approximately 1 molar, then the absorbance will be on the order of 0.2. For our system, this would correspond to a reference transmission of 2500 counts and a sample transmission of 1600 counts. For comparison, published absorption spectra of carbon disulfide² and nitromethane¹⁸ are shown in Figures 22 and 23.

4.0 ERROR ANALYSIS

It is necessary to assign a confidence level to the results thus requiring a knowledge of the system's errors. Some systematic errors can be corrected (i.e. reflection losses at sapphire-liquid interfaces). Others, such as attenuation by the various optical components, are not easily compensated. Since these errors are reproducible, their effect is constant and will not affect the accuracy of the results.

Random errors cause fluctuations in observations so results differ between any two or more experiments. Random errors are not reproducible. To reduce random errors it is necessary to improve the experiment and refine the technique as much as possible. The random errors in the absorption experiment are:

- (1) Light induced background in the streak camera. For this particular camera, the light induced background has been estimated to be 8% of the average intensity². As long as intensities are high, the effect of this error is small. For shot data, calculating only absorbance changes will eliminate this error.
- (2) Pulse-to-pulse flashlamp intensity fluctuations. For the Xenon Corporation lamp, it is estimated that the maximum error in intensity is 8-10%. This error is reduced to 1-2% by consistently flashing the lamp. Also, this error is completely eliminated for shot data if early (pre-shockup but post-impact) tracks are used as references for the later tracks.
- (3) Nonlinearity of intensity input/output for the streak camera/OMA system². This error is less than 0.6%.

- (4) Errors from assuming normal incidence in calculations². These are less than 2.5%.
- (5) Errors from the counting statistics in the vidicon. These errors are typically taken as the square root of the counts, i.e., the reference signal = $\text{Ref} \pm \sqrt{\text{Ref}}$. They are usually < 5-8%.
- (6) Stray light entering the system. It is very difficult to assigned a percentage to this error.

The smaller the random errors, the more precise the experiment. For solutions with a large absorbance, stray light has the greatest effect on the results. Low absorbance solutions are most affected by instrumental variables such as the counting efficiency of the vidicon.

The errors or uncertainty of the measurements have to be mathematically propagated through the absorbance calculation. This is shown in Appendix D. An important fact is even with (at least) six random errors, they do not add together to give a 'total' error. Their very nature of being random says the largest error will dominate. If however, the errors are approximately the same, then they will add by the root-mean-square¹⁹. Thus the errors listed above will combine as:

$$\begin{aligned} \text{Error} &= \sqrt{(0.08)^2 + (0.02)^2 + (0.006)^2 + (0.025)^2 + (0.08)^2} \\ &= .11 \\ &\approx 10\% \end{aligned}$$

The largest random counting error is on the order of 10% and this error is propagated into the absorbance error in Appendix D. The results of Appendix D show the largest absorbance error is $\approx 6\%$. Figure 24 is Figure 21 (an absorption spectrum of nitromethane) but with error bars added.

5.0 REFERENCES

1. R.L. Webb, MS thesis, Washington State University, 1990.
2. R.L. Gustavsen, Ph.D. Dissertation, Washington State University, 1989.
3. C.S. Yoo and Y.M. Gupta, J. Phys. Chem. 94, 2857 (1990).
4. Xenon Corporation, Woburn, MA 01801.
5. The off-axis parabolic mirror is an 'unknown' component. According to Dr. C.S. Yoo, the mirror was 'on-hand' when he decided to use it for turning the flashlamp beam. He does not know the manufacturer or if there is any special UV enhancement coating on it. He recommended caution on its use below 350nm. However; Mr. Richard Webb claims that it was bought from Melles Griot with a special UV enhancement coating applied. Neither claim has been verified.
6. Melles Griot, Irvine, CA. 92714.
7. Technical Services, Washington State University.
8. Union Carbide, San Diego, CA. 92123.
9. Oriel Corporation, Stratford, CT. 06497.
10. Mitsubishi Cable America, Inc. New York, NY 10022.
11. SPEX Industries, Inc., Edison, NJ 08820.
12. Cordin Co., Salt Lake City, UT 84119.
13. EG&G Princeton Applied Research, Princeton, NJ 08540.
14. Hewlett-Packard,
15. David Richardson Grating Laboratory, Analytical Products Division, Milton Roy Company, Rochester, NY 14625 (note: Milton Roy Co. now owns the grating division formerly owned by Bausch and Lomb).
16. Hamamatsu Corporation, Middlesex, NJ 08846.
17. Mr. Lou Panico (pronounced ' PA-KNEE-KO'), Xenon Corporation, (800) 878-3594.
18. *UV Atlas of Organic Compounds*, Vol. III, Plenum Press, New York (1967).
19. W. Lichten, *Data and Error Analysis*, Simon & Schuster, Newton MA (1988).
20. Melles Griot Catalog (1991).
21. E. Hecht, *Optics*, Addison-Wesley Publishing Co., Reading MA (1987).
22. *International Critical Tables*, Vol. VII, McGraw Hill, New York (1930)
23. G.H. Schenk, *Absorption of Light and Ultraviolet Radiation: Fluorescence and Phosphorescence Emission*, Allyn and Bacon, Inc., Boston (1973).
24. E.J. Meehan, *Optical Methods of Analysis*, Intersciences Publishers, New York

24. E.J. Meehan, *Optical Methods of Analysis*, Intersciences Publishers, New York (1966).
25. A.A. Schilt and B. Jaselskis, in *Treatise on Analytical Chemistry: Part I Vol. 5*, John Wiley & Sons, Inc., Easton PA (1964).
26. G.H. Ayres, *Anal. Chem.* 21, 652 (1949).
27. F. Twyman and G.F. Lothian, *Proc. Phys. Soc. (London)* 45, 643 (1933).
28. L.S. Goldring, R.C. Hawes, G.H. Hare, A.O. Beckman, and M.E. Stickney, *Anal. Chem.* 25, 869 (1953).
29. C.F. Hiskey, *Anal. Chem.* 21, 1440 (1949).
30. P.R. Bevington, *Data Reduction and Error Analysis for the Physical Sciences*, Case Western Reserve University (1969).

Table 1: Components

Component / Reference	Manufacturer	Model #	Description
Flashlamp (4)	Xenon Corporation	457 Micropulser	Xenon Flashlamp
Parabolic mirror (6)	Melles Griot	02POA017/028	*
Turning mirror (6)	Melles Griot	01MFG011	Front Surface, 50x50 mm with 3 mm thick, with #028 UV coating
Turning mirrors (2) on projectile (6)	Melles Griot	01MFG005	Front Surface, 12.5x12.5 mm with 1 mm thick, with #028 UV coating
Aperture (7)	Technical Services, WSU	NA	Brass washer, 1" diameter with 0.25" centered hole
Lens #1 (9)	Oriel Corp.	41230	Fused silica plano convex, 12.7mm dia., 38 mm FL
Lens #2 (9)	Oriel Corp.	41209	Fused silica plano convex, 12.7mm dia., 16 mm FL
Optical Fiber (10)	Mitsubishi Cable Industries, Ltd.	STU400E-SY	UV transmission fiber, 400 micron diameter
Spectrometer (11)	SPEX Industries	Minimate-2	220mm FL, f/4 aperture
Diffraction Grating (15)	Milton Roy Co.	35-33-10-090	300 grooves/mm, blaze wavelength 300nm
Streak Camera (12)	Cordin Co.	Cordin 160 Model #5B	Image converter camera
Vidicon Detector (13)	EG&G PAR	1252B	Silicon diode array 512x512 pixels
Multichannel Detector Controller (13)	EG&G PAR	1216	Detector controller
Optical Multichannel Analyzer (13)	EG&G PAR	OMA III	Sets parameters for controller and records data
Computer (14)	Hewlett-Packard	HP 9000	

* Note: The off-axis parabolic mirror is an 'unknown' component. According to Dr. C.S. Yoo, the mirror was 'on-hand' when he decided to use it for turning the flashlamp beam. He does not know the manufacturer or if there is any special UV enhancement coating on it. He recommended caution on its use below 350nm. However; Mr. Richard Webb claims that it was bought from Melles Griot with a special UV enhancement coating applied. I have been unable to verify either claim.

Table 2: Spectral Efficiencies

Wavelength (nm)	300	400	500
Sapphire Impactor (12.7mm thick)	0.731	0.777	0.808
Sapphire Liquid Cell - Front Window (3.175mm thick)	0.817	0.834	0.844
Sapphire Liquid Cell - Back Window (12.7mm thick)	0.731	0.777	0.808
Lens #1 (2.2mm thick)	0.925	0.930	0.930
Lens #2 (4.7mm thick)	0.923	0.930	0.930
Fiber #1	0.901	0.923	0.928
Fiber #2	0.813	0.897	0.918
Spectrometer Mirror #1	0.880	0.920	0.920
Spectrometer Mirror #2	0.880	0.920	0.920
Diffraction Grating	0.660	0.522	0.386
Spectrometer Mirror #3	0.880	0.920	0.920
Fused Silica Face Plate (Streak Camera)	0.850	0.860	0.871
Streak Camera Photocathode	0.520	0.950	0.800
TOTAL EFFICIENCY	0.054	0.120	0.085

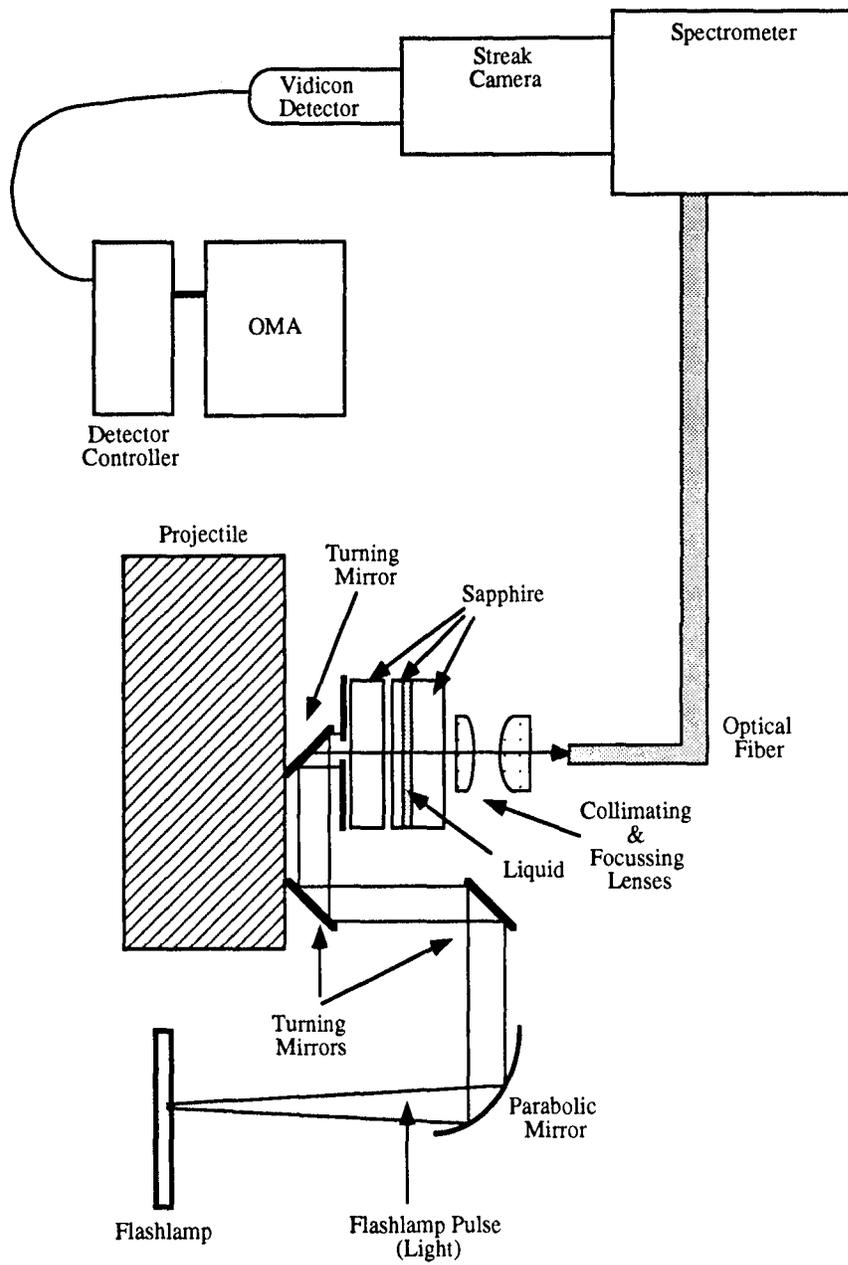
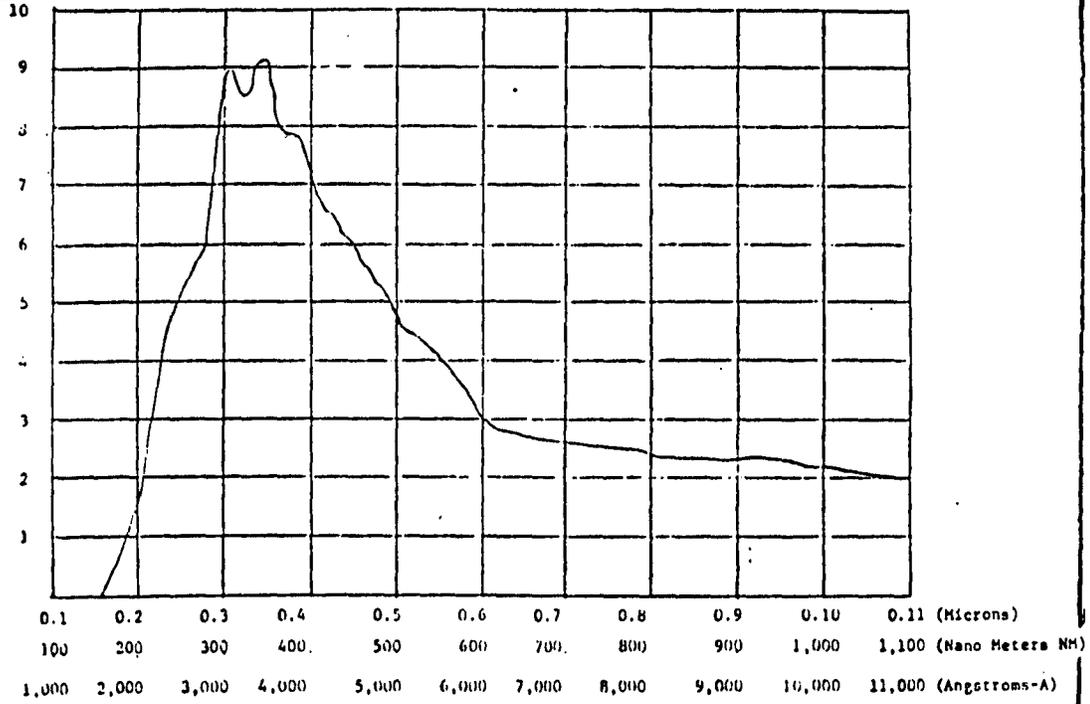


Figure 1: Schematic view of experimental configuration.

General Spectral Distribution Of Xenon Used High Energy Flashtubes *See below

Xenon

Relative Units



IR ————

XENON Corporation
 a General Spectral Distribution Of
 Micro Pulse Flashtubes At 10 KV Anode Voltage

Figure 2: Spectral Distribution Of Micropulse Xenon Flashtube (10kV). From reference #4.

ULTRAVIOLET-ENHANCED ALUMINUM

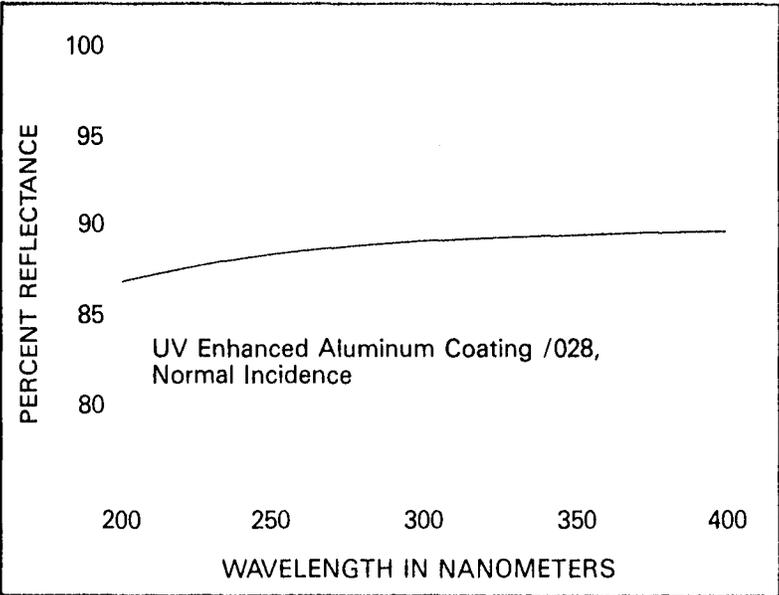


Figure 3: Reflectance of UV enhanced mirrors. From reference #6.

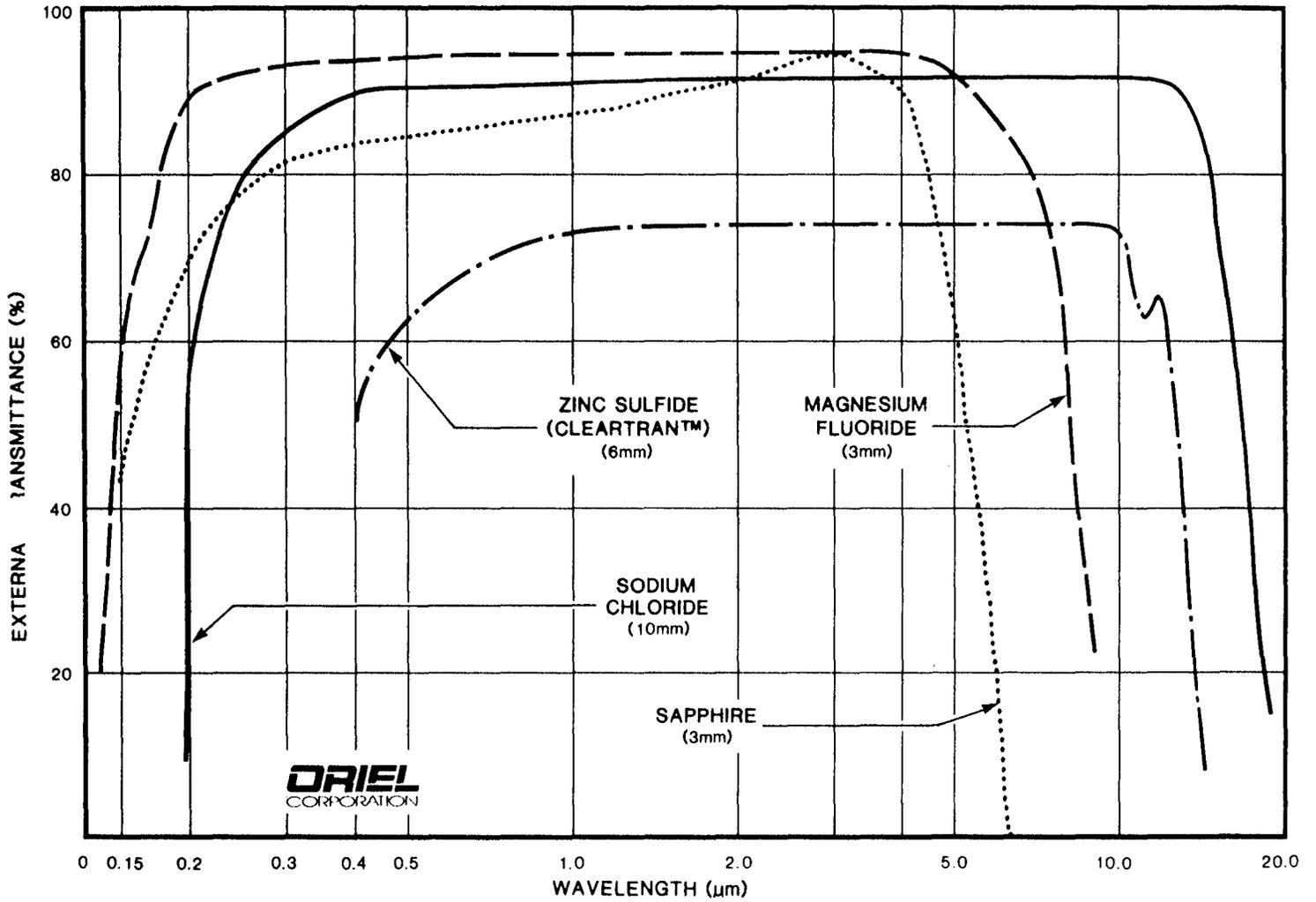


Figure 4: External Transmittance of sapphire. (The transmittances shown include losses from two surface reflections and the material's absorption). From reference #9.

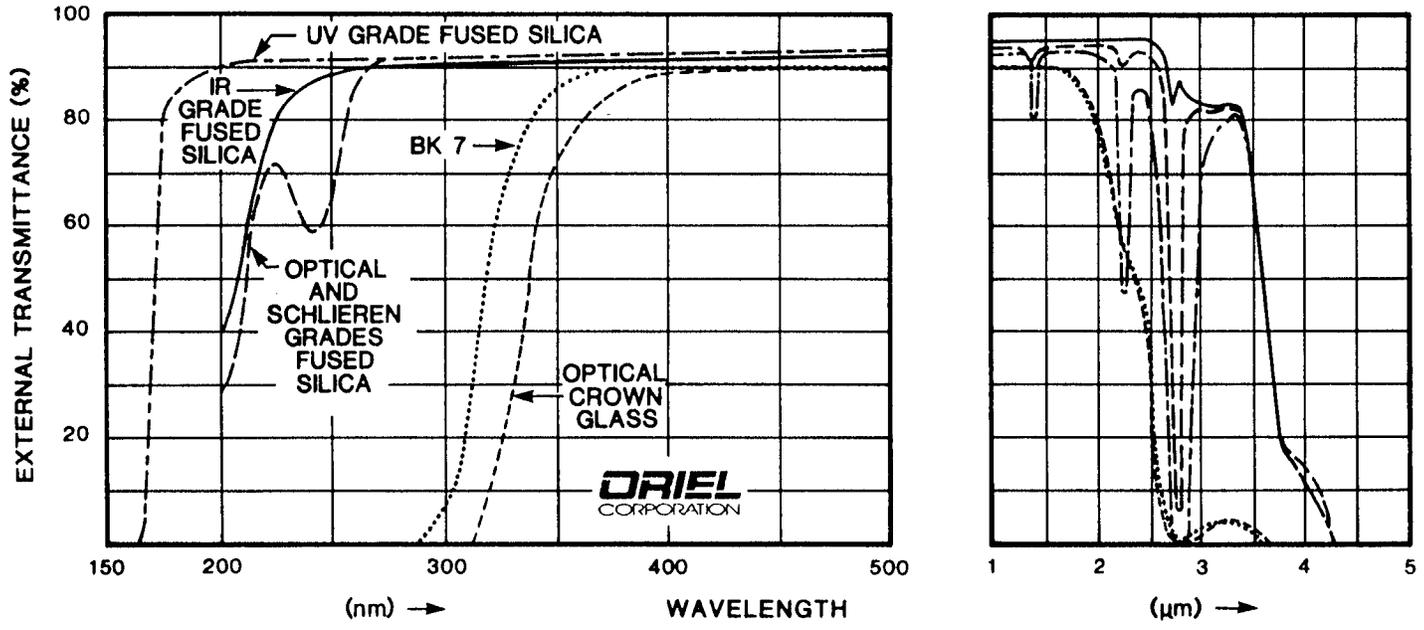


Figure 5: External Transmittance of UV grade fused silica. (The transmittances shown include losses from two surface reflections and the material's absorption).

From reference #9.

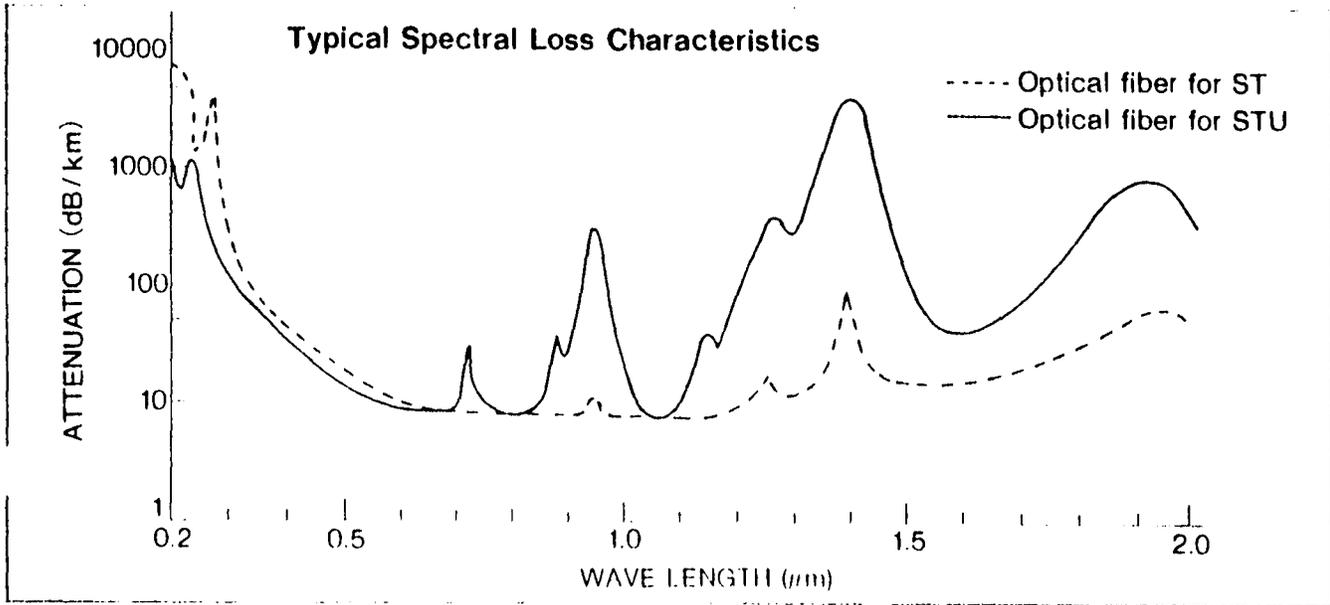


Figure 6: Attenuation of optical fiber. From reference #10.

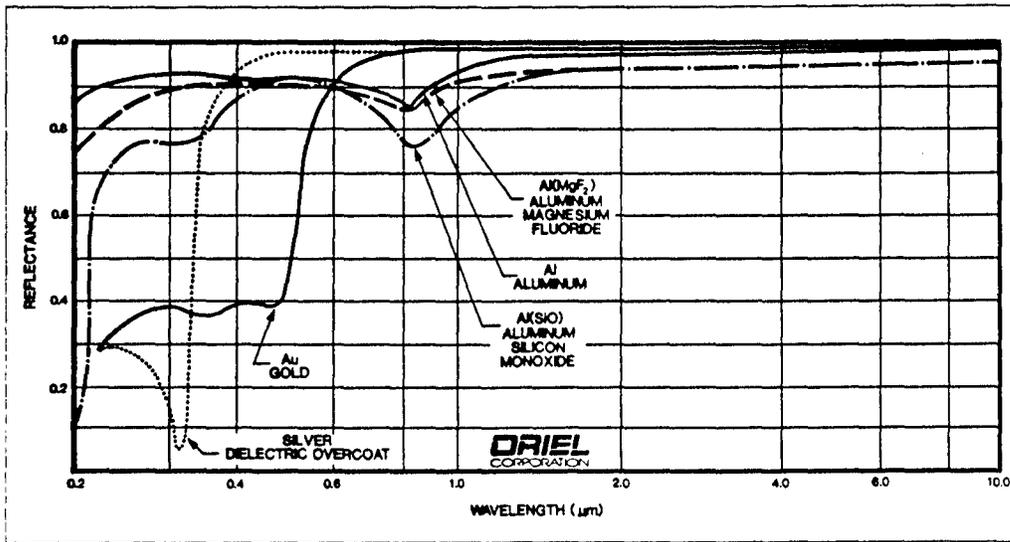
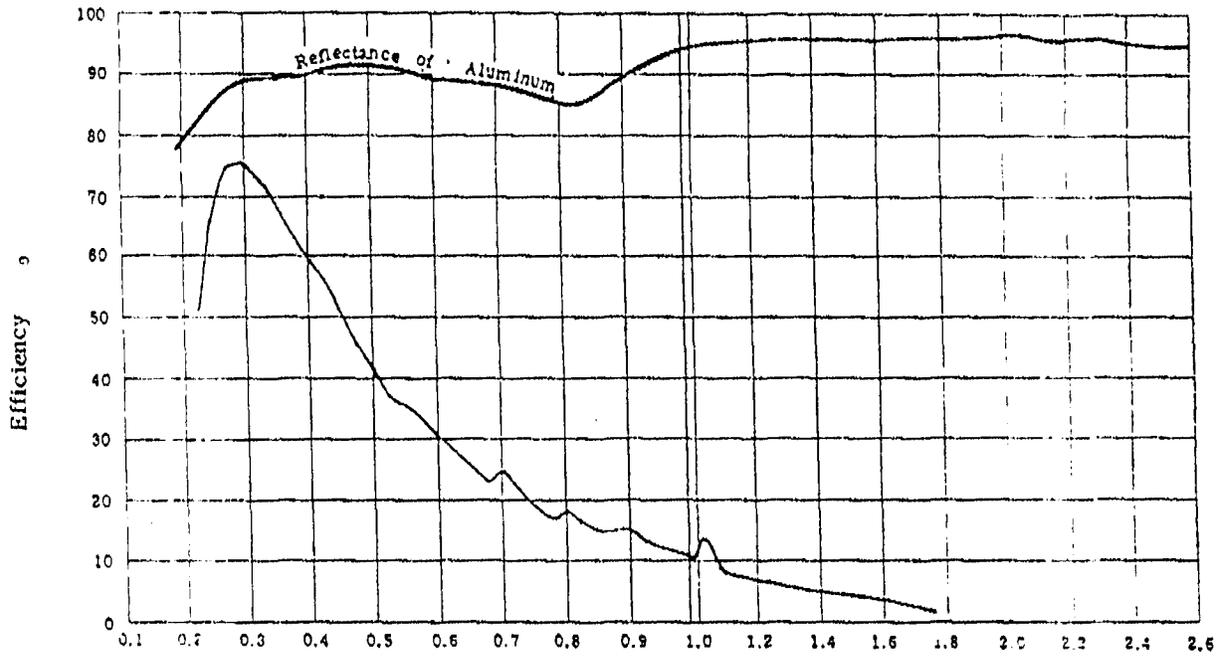


Figure 7: Reflectance of aluminum magnesium fluoride mirrors. From reference #9.

DIFFRACTION GRATING EFFICIENCY

Measured under near-Littrow conditions
with 8° between incident and diffracted beams
- relative to reflectance of aluminum

Polarized at 45°



Wavelength - μm

1st Order

Serial No. 1129-1 Date 6-25-73
 Grooves/mm 300 Blaze Angle $2^\circ 35'$
 Type Reflection Blaze Wavelength 3000 μ
 Remarks Cat. #35-53*-090
35-33-10-09002 x128

Figure 8: Diffraction grating efficiency. From reference #15.

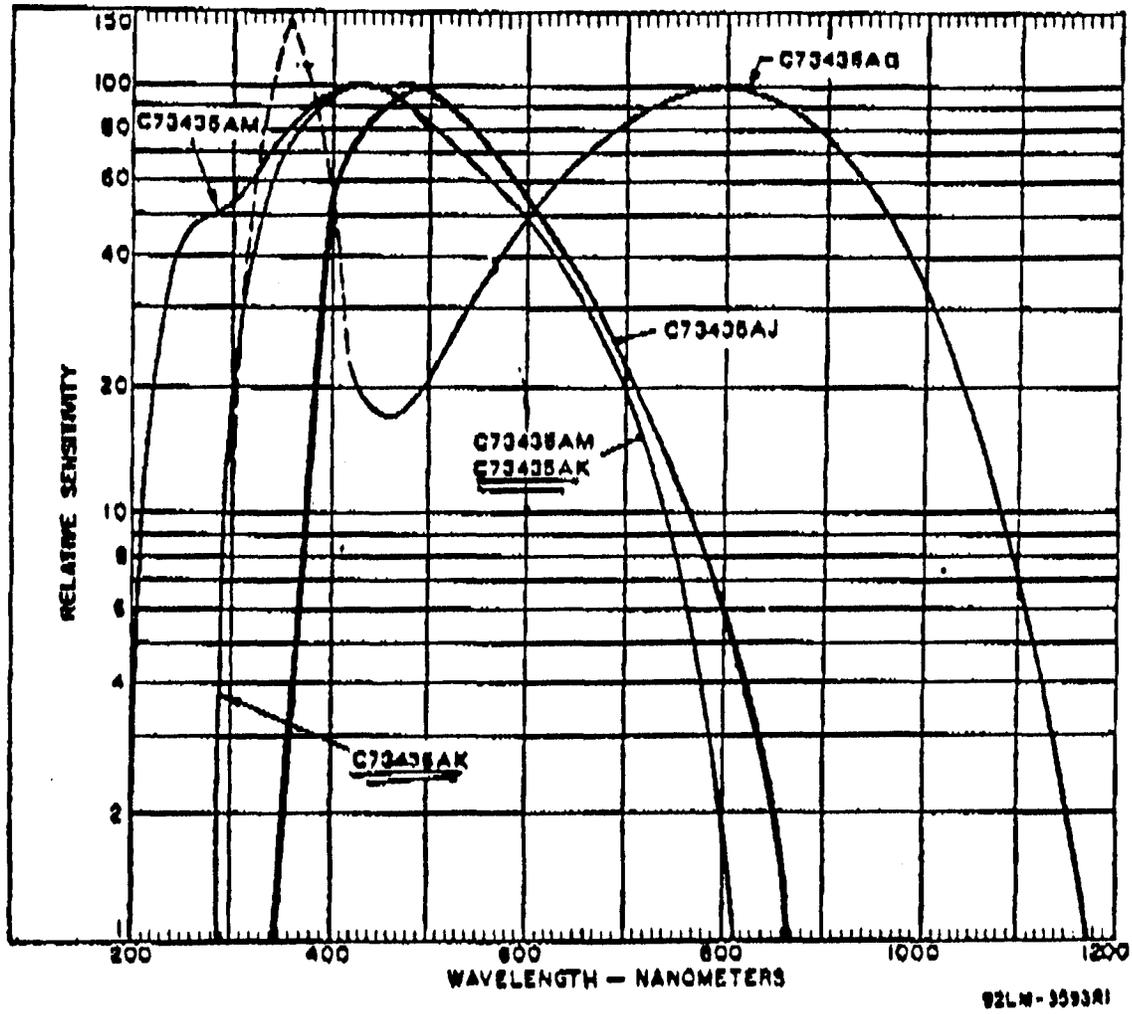


Figure 9: Relative sensitivity of the streak camera. The photocathode used is # C73435Y. From reference #12.

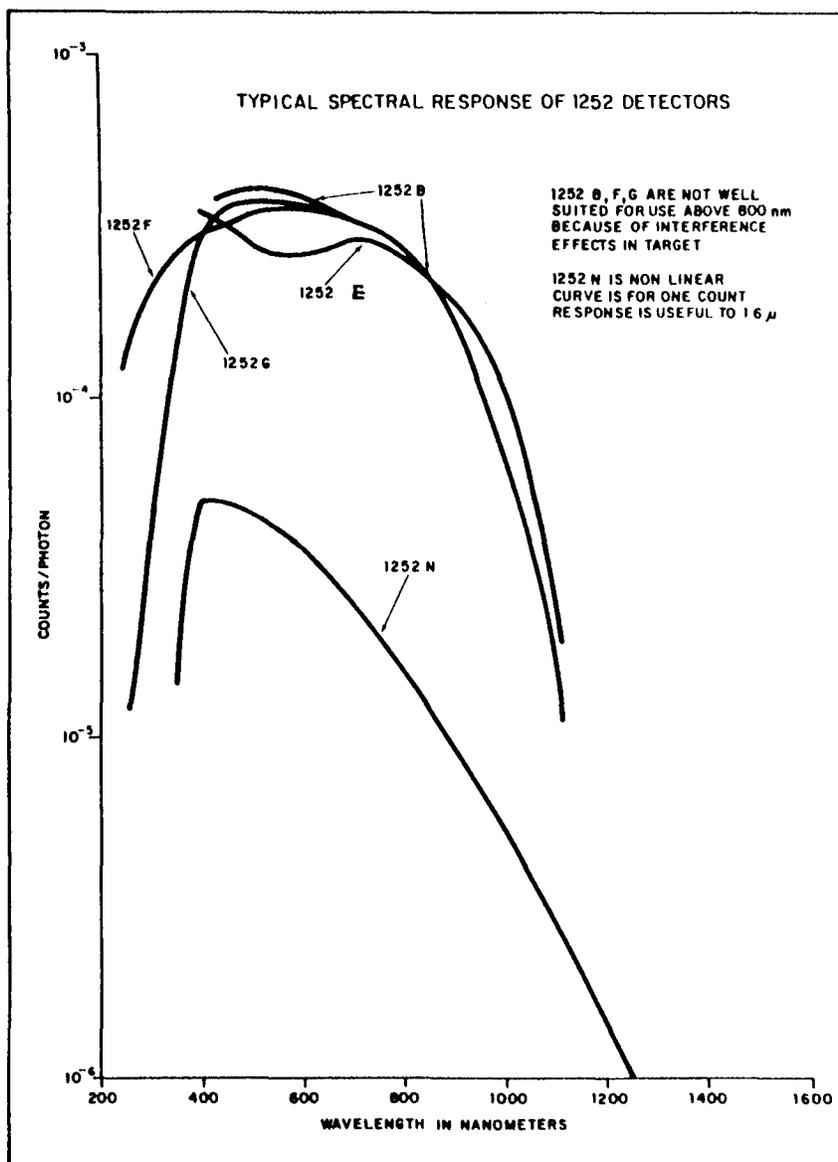


Figure 10: Spectral Response of Vidicon Detector. From reference #13.

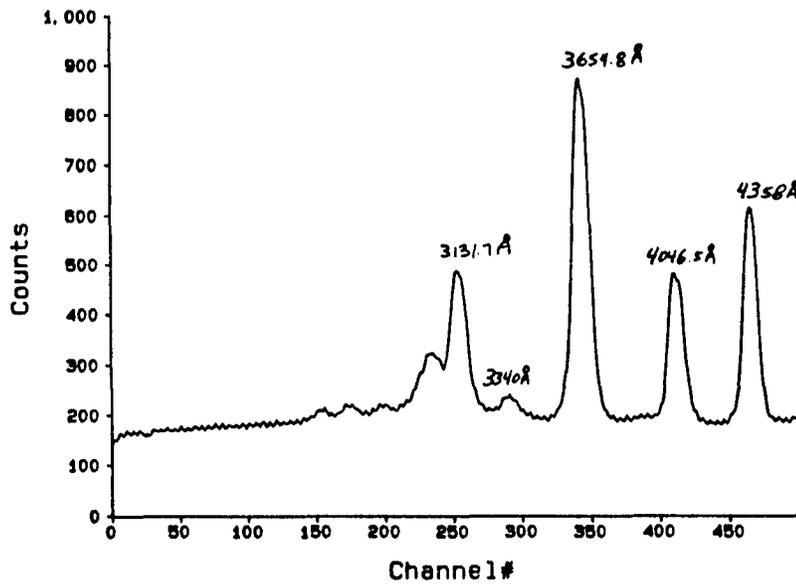
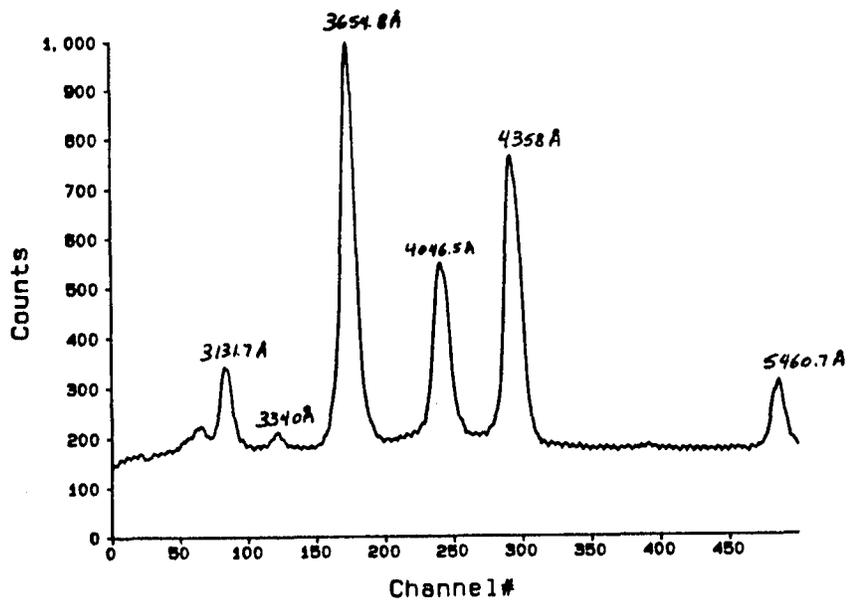


Figure 11: The mercury spectrum of the 'old' lamp. The top spectrum (grating set at 0150) has a spectral range of 2650-5550 angstroms. The bottom spectrum's (grating set at 0125) range is 1700-4560 angstroms.

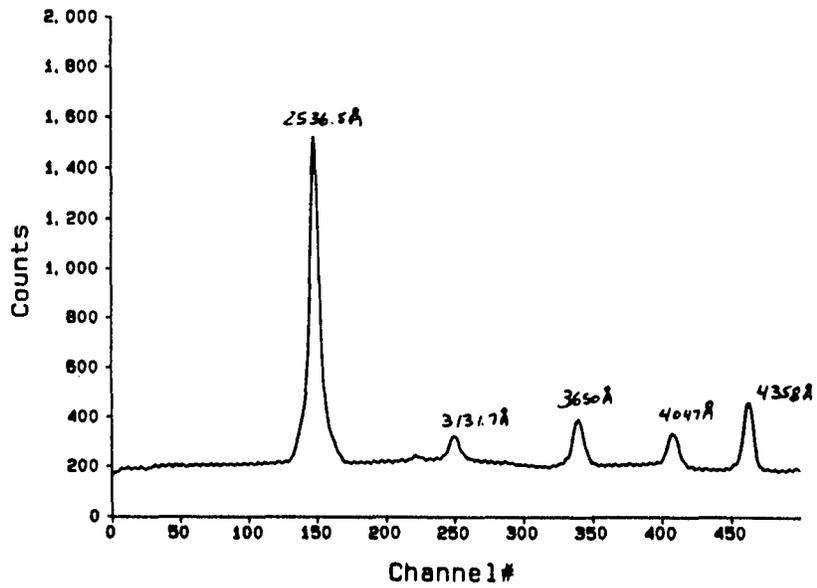
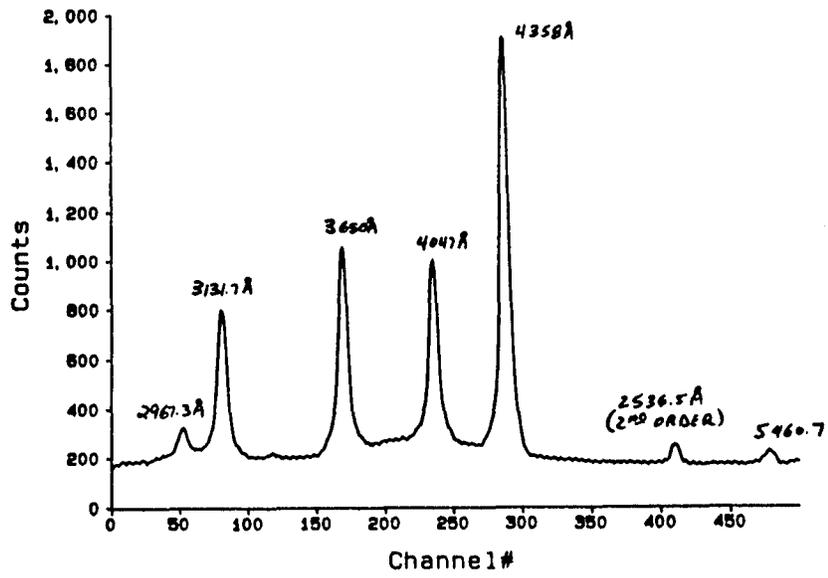
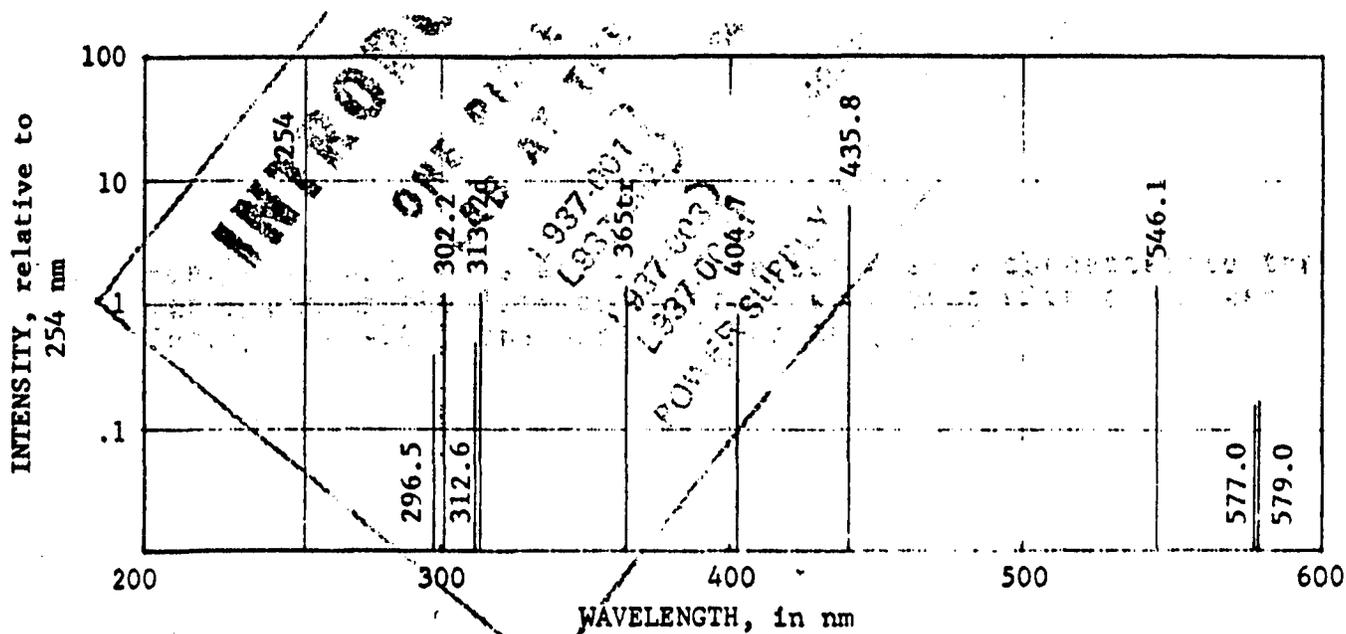


Figure 12: The mercury spectrum of Dr. K. Hipps' lamp. The top spectrum (grating set at 0150) has a spectral range of 2650-5550 angstroms. The bottom spectrum's (grating set at 0125) range is 1700-4560 angstroms.



*Vycor is a registered trademark of the Corning Corporation

Figure 13: The calibrated mercury spectrum of Dr. K. Hippi's lamp. From reference #16.

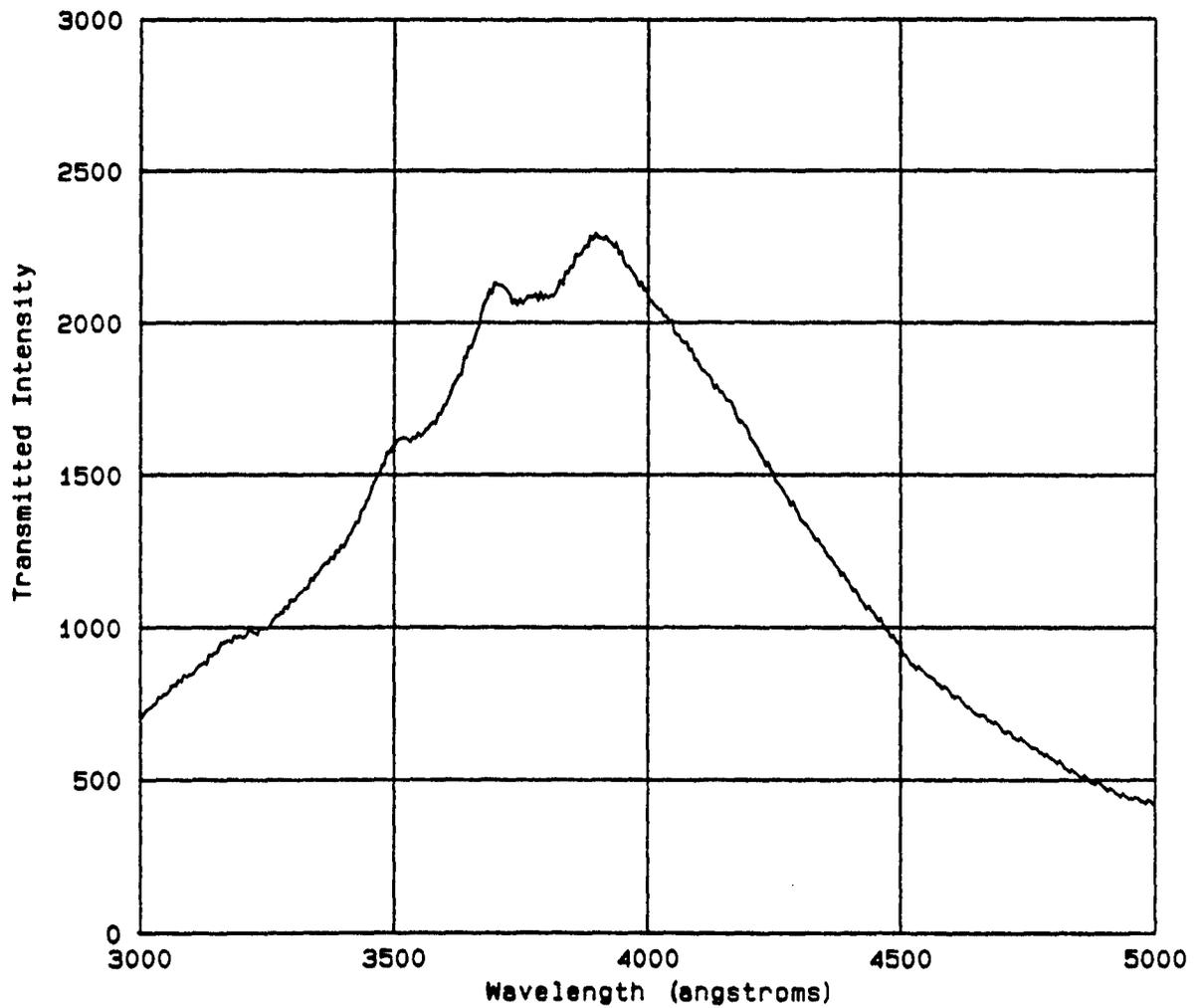


Figure 14: The spectral response of the xenon flashtube, as recorded by the absorption spectroscopy system, of shot 92-006. This is track #25 of the air-filled, ambient cell. This grating used was blazed for 300nm.

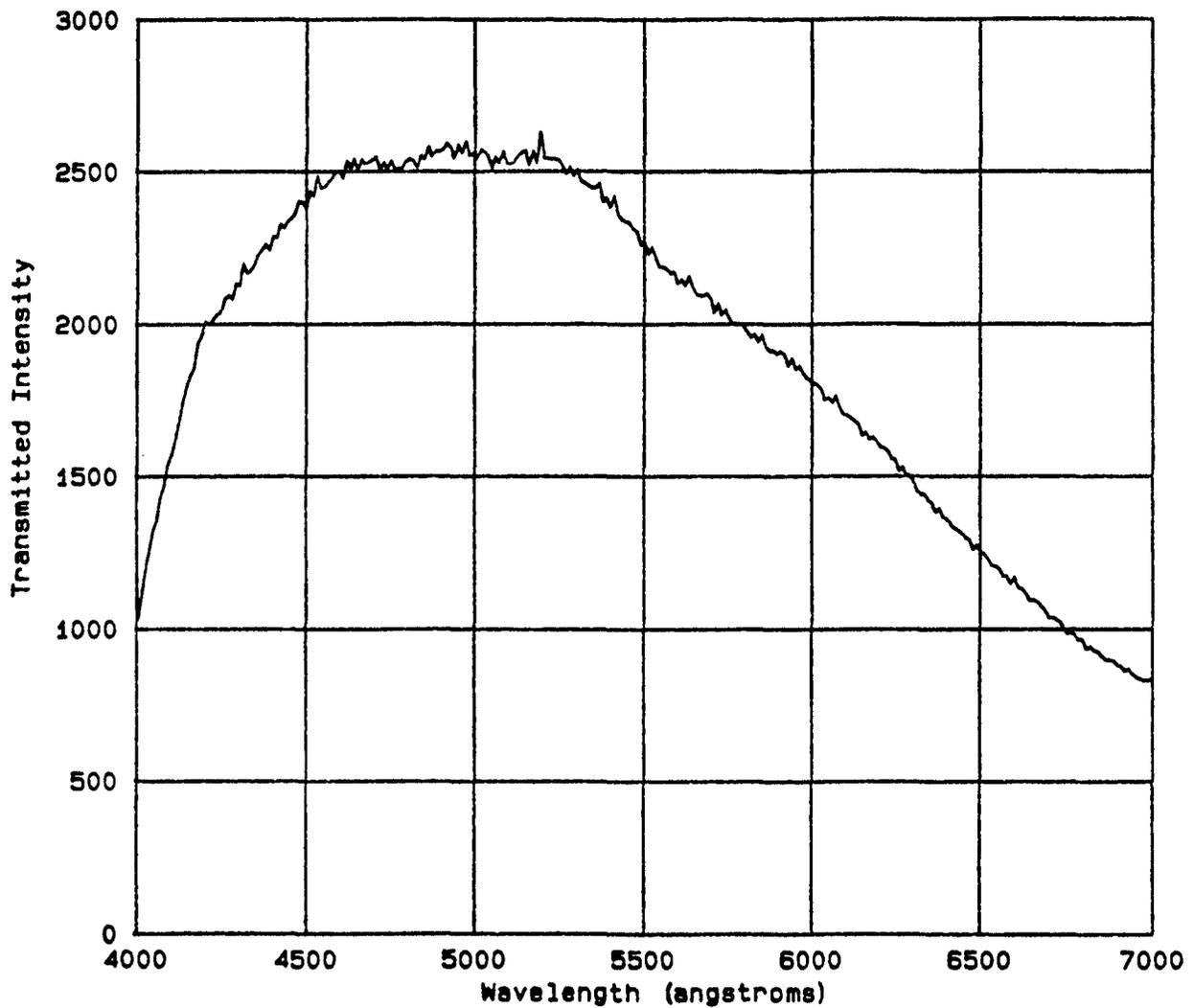


Figure 15: The spectral response of the xenon flashtube, as recorded by the absorption spectroscopy system. This is track #25 of the air-filled, ambient cell. This grating used was blazed for 500nm.

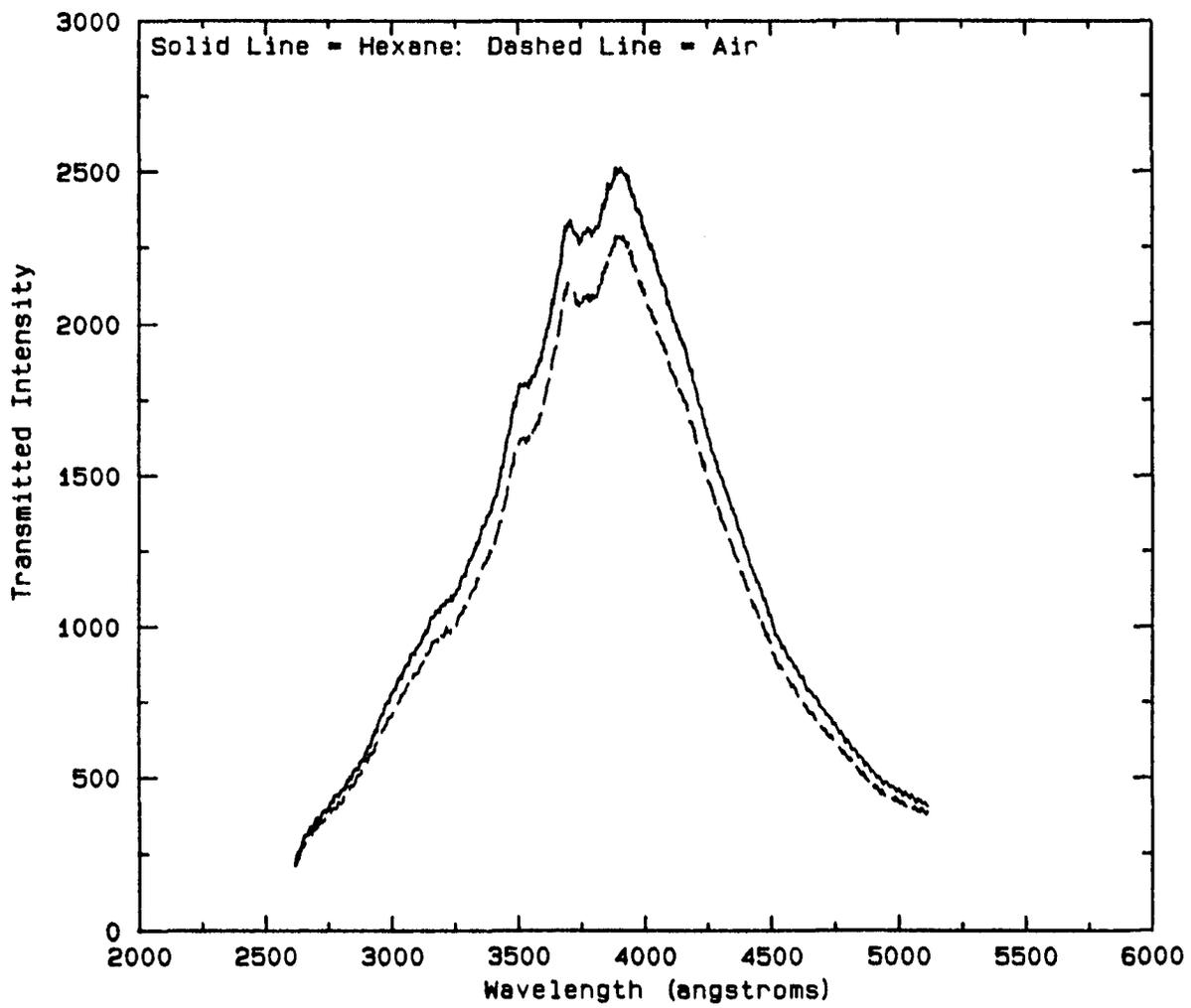


Figure 16: The spectral response, as recorded by the absorption spectroscopy system, of shot 92-006. This is track #10 of the hexane-filled, ambient cell.

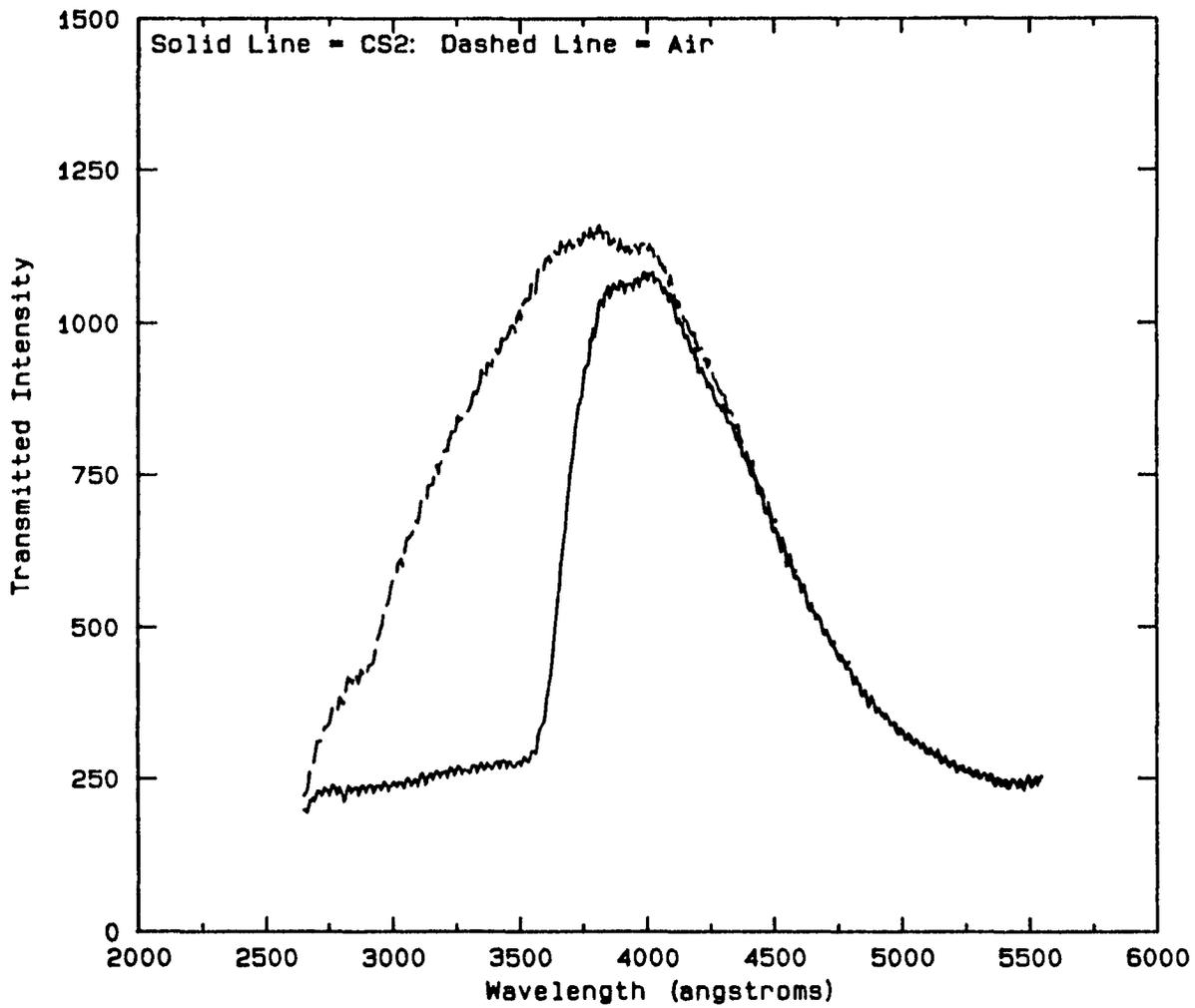


Figure 17: The spectral response, as recorded by the absorption spectroscopy system on 1/25/92. This is track #20 of the carbon disulfide-filled, ambient cell.

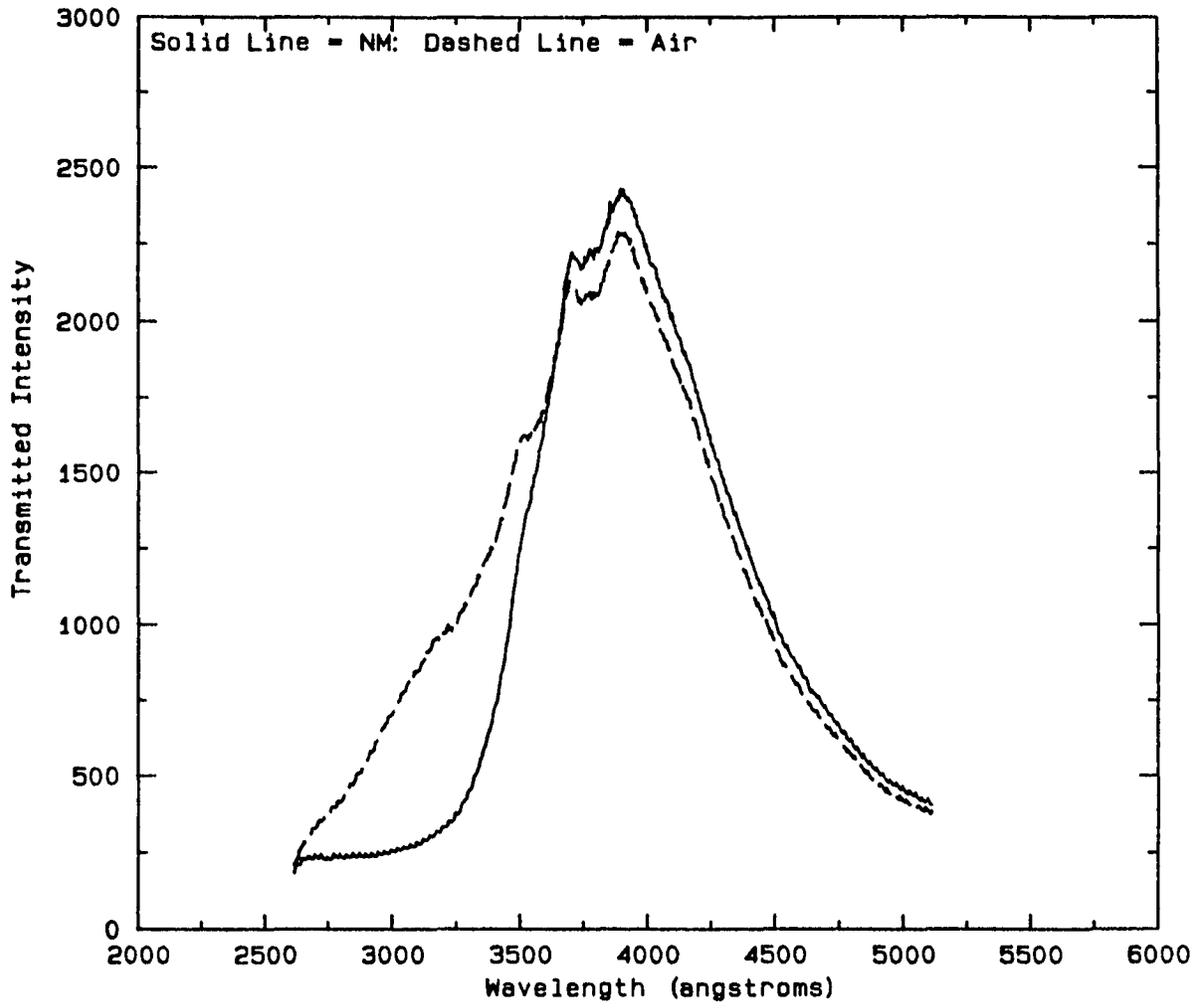


Figure 18: The spectral response, as recorded by the absorption spectroscopy system, of shot 92-006. This is track #10 of the nitromethane-filled, ambient cell.

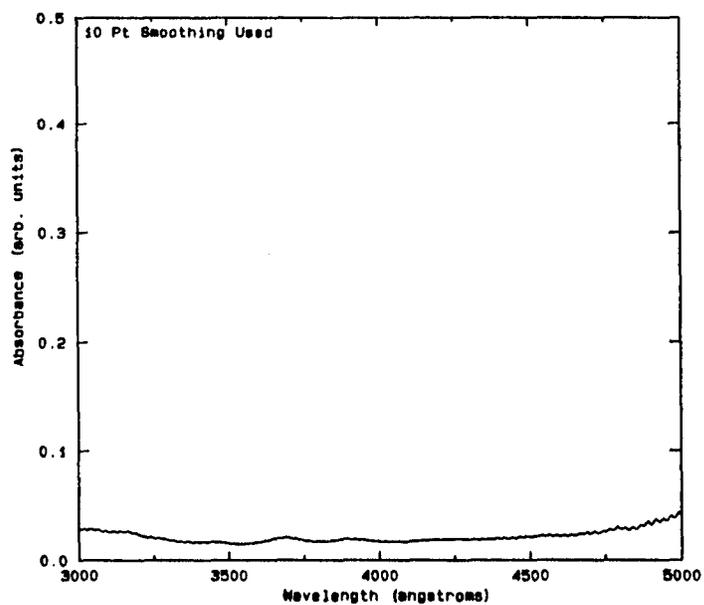
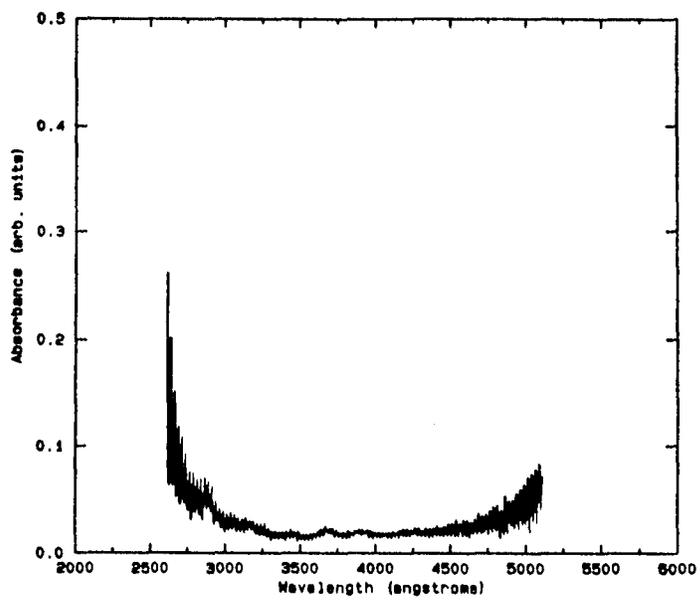


Figure 19: Absorption spectrum of hexane-filled cell. The top plot is the unsmoothed, all wavelengths data. The bottom plot is smoothed.

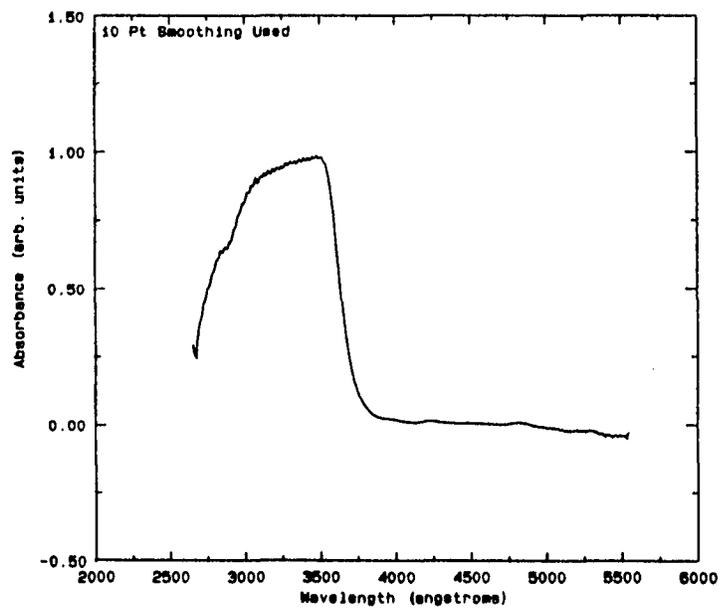
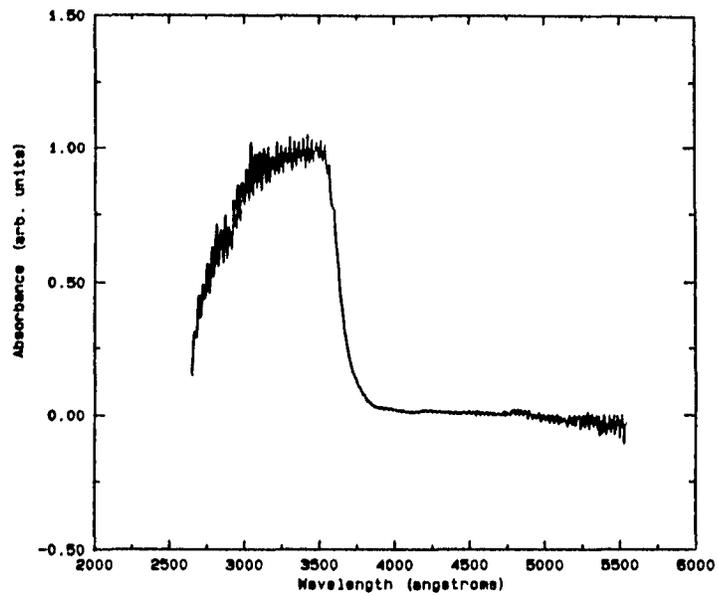


Figure 20: Absorption spectrum of carbon disulfide-filled cell. The top plot is the unsmoothed and the bottom is smoothed.

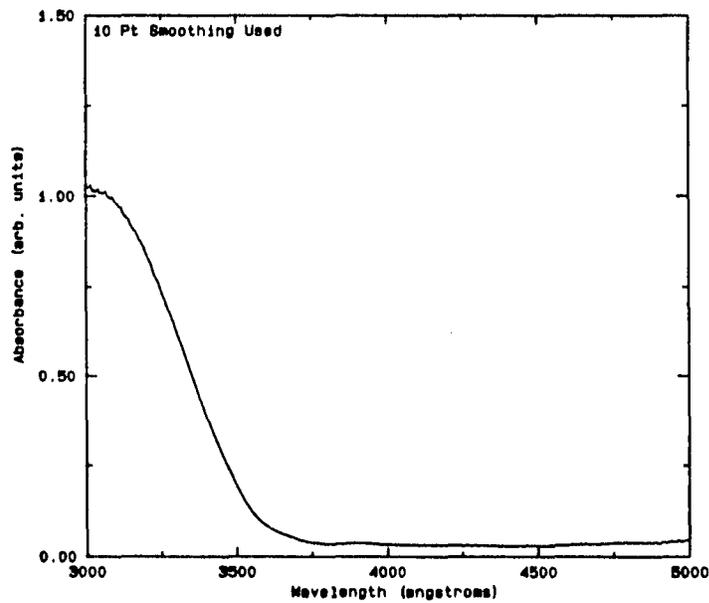
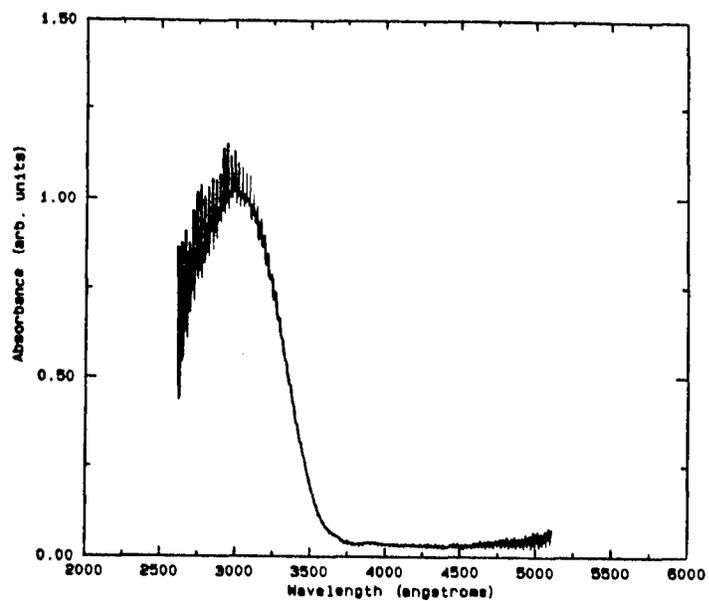


Figure 21: Absorption spectrum of nitromethane-filled cell. The top plot is the unsmoothed and the bottom is smoothed.

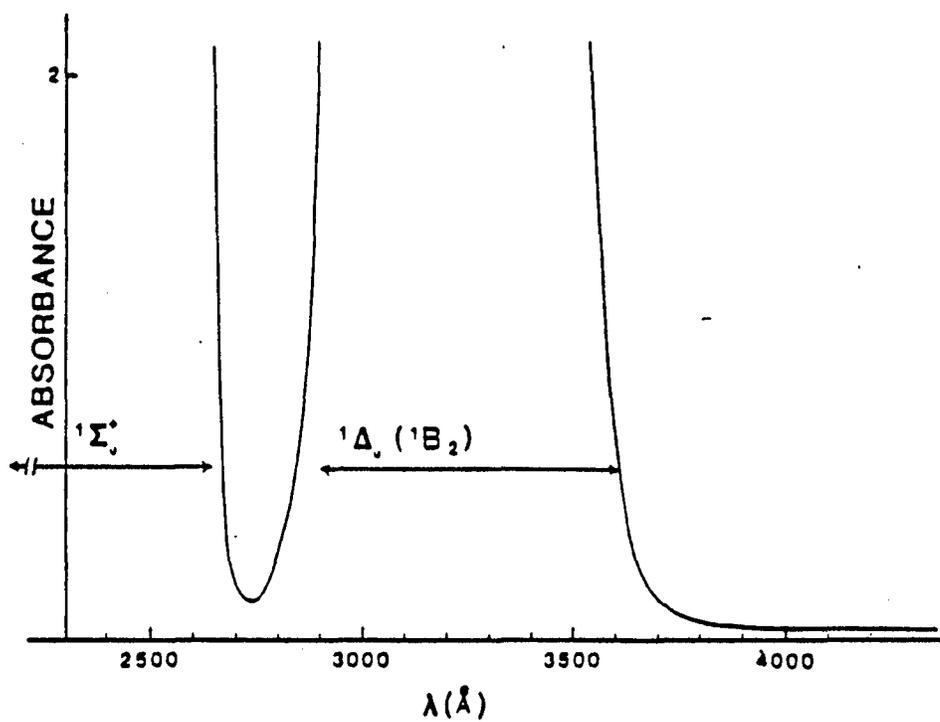


Figure 22: Published absorption spectra of carbon disulfide. From reference #2.

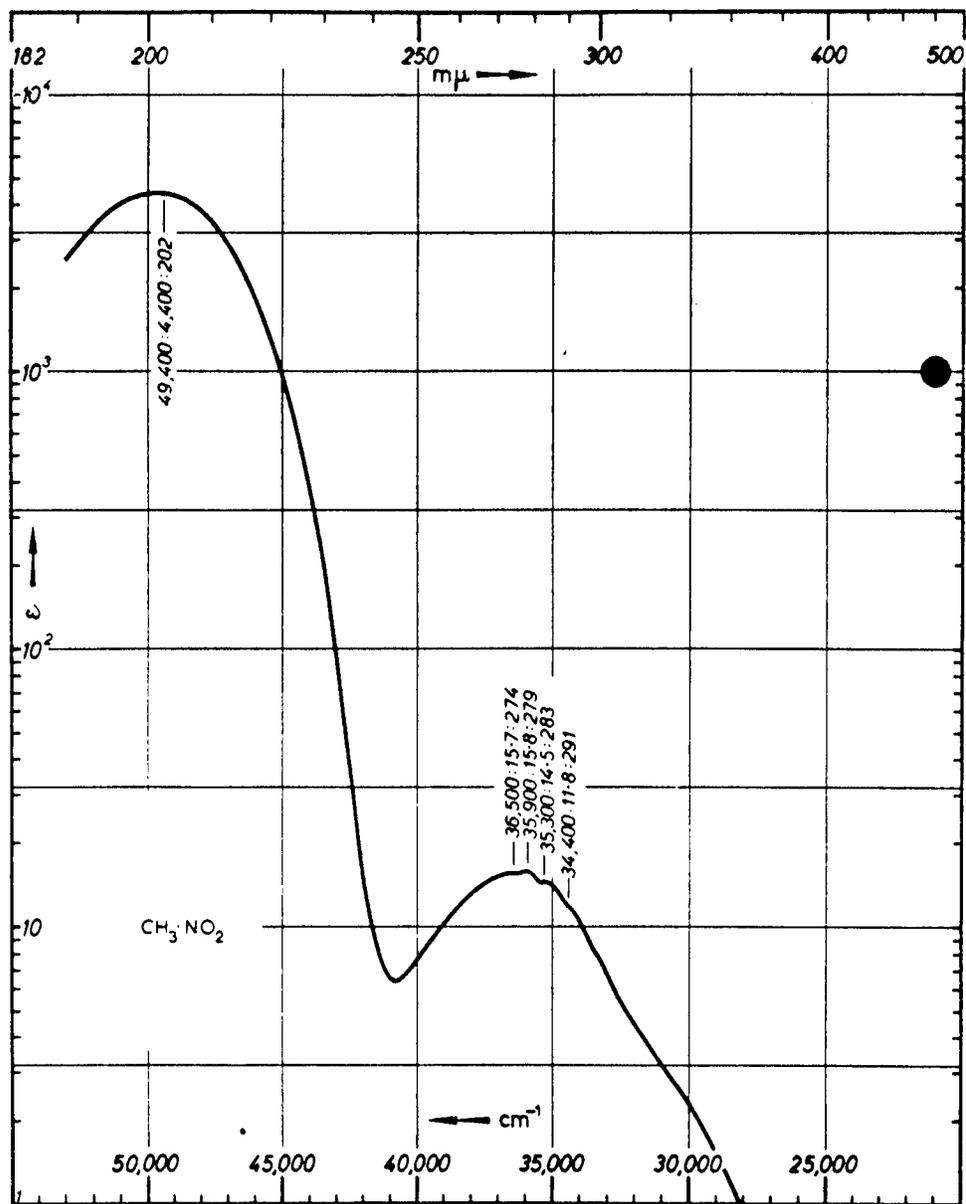


Figure 23: Published absorption spectra of nitromethane. From reference #18.

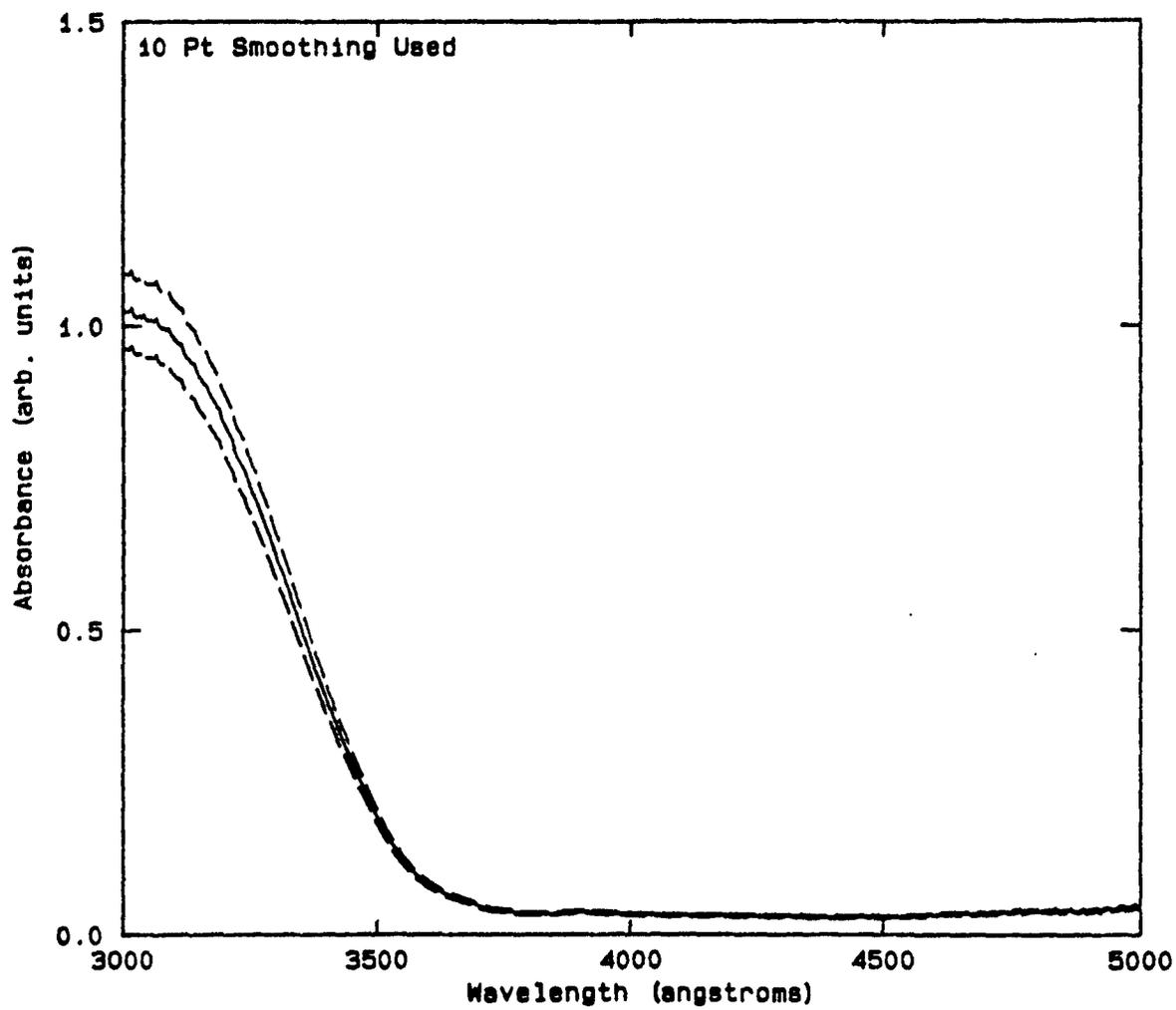


Figure 24: Absorption spectrum of nitromethane-filled cell with error bars. The error shown is the maximum error of 6%.

Appendix A: Lens Collection System

Modifying the collection system to use an optical fiber made it crucial to have efficient photon collection. Having a fiber end collecting photons would only be 0.4% efficient (the ratio of the areas of the 0.25" aperture and the 400 micron fiber diameter). Going from a 6.35mm diameter circle (0.25") to a 0.400 mm diameter (400 microns) is a magnification of $\frac{1}{16}$. Initially a single lens was considered.

The theoretical spot size a lens will form is known as the Airy disc and is calculated via²⁰:

$$D = 2.44 * \text{Wavelength} * \frac{f}{\#} \quad \text{A.1}$$

The $\frac{f}{\#}$ of a lens is the ratio of the focal length to the lens diameter. Ideally, the spot size should be approximately 1000-1500 microns in diameter. This solves two problems simultaneously: discarding photons to avoid detector saturation and a guarantee of coupling the light into the fiber.

To achieve a 1000 microns spot size and still use a 0.5" diameter lens, a focal length of 9 meters (for 600nm light) is needed. This is impractical. An alternate possibility is to defocus the spot which would allow use of shorter focal length lenses. For instance, using a 38mm focal length lens the spot size would be 1000 microns 44mm away from the lens. The problem of using a single lens is assumption of completely collimated, transmitted light.

The off-axis, parabolic turning mirror (see Figure 1) supposedly collimates the beam. However, if the mirror is even slightly misaligned the collimation is incomplete. For this reason, a two lens system was designed. Utilizing a second lens will ensure complete collimation and the loss of photons is still minimal. Figure A.1 shows how the lenses are placed.

The first lens, focal length of 38mm, gathers the light and collimates it. The collimated light is focused by the second lens, focal length of 16mm, onto the fiber head. To be able to collect all the photons coming through the 0.25" aperture, the first lens must be placed approximately 3" away from the aperture. Doing so will ensure maximum collection. The second lens 'sees' collimated light (i.e., the object distance is infinity) and will focus the image at the focal point. See Figure A.2. Placing the fiber either 12.14mm or 19.86mm away from this second lens will defocus the image. These are very precise distances which

in practice might be difficult to set exactly. The lens and fiber holder were designed with some leeway allowing the researcher to 'fine-tune' the distance between the first lens and the sample and the distance between the second lens and the fiber end.

One other possibility tested was using a single lens but imaging the flashtube onto the fiber end. This would make the object distance be approximately 1800mm (6'). Using the 38mm focal length lens, the image distance is then 38.82mm. The magnification is $0.021 \left(\frac{38.82}{1800} \right)$ resulting in a spot size of 140 microns. It is possible to defocus (move the fiber back 3mm) and achieve a 1000 micron spot size. For shot 92-006 a single lens system and a two lens system were tested. The double lens system gave about 30% more counts and it was for this reason the two lens system was used.

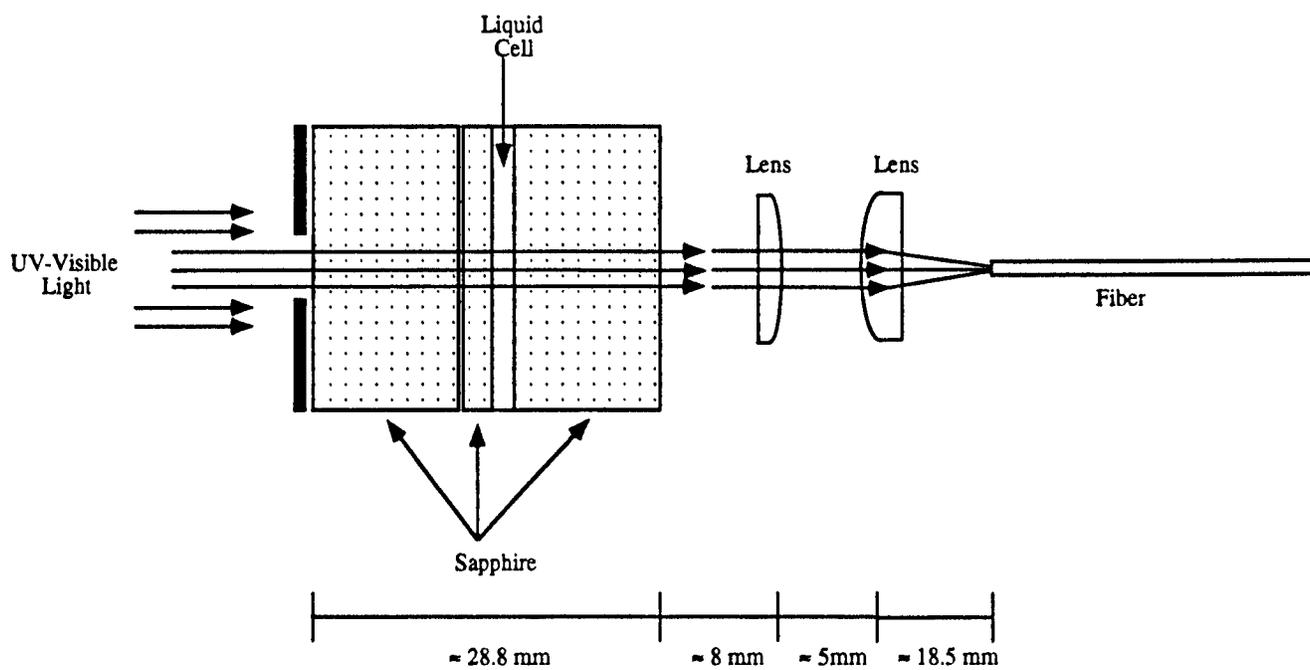
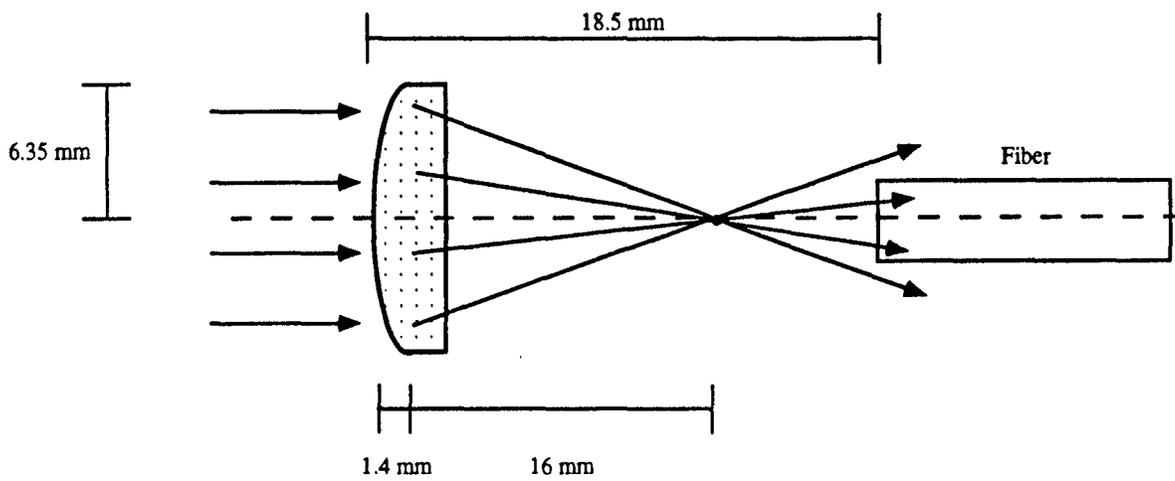


Figure A.1: Schematic view of lens configuration.



$$\tan(\theta) = 6.35 / 16 \text{ or } \theta = 21.6^\circ$$

$$\tan(21.6^\circ) = Y / X = Y / (18.5 - 1.4 - 16) = Y / 1.1$$

$$Y = 0.436 \text{ mm}$$

The total spot size is $2 * 0.436 \text{ mm} * 1000 = 870 \text{ microns}$

Figure A.2: Focussing the light rays onto the fiber end.

Appendix B: Reflection Losses

Figure B.1 is a diagram of a ray passing through three mediums (two interfaces). This is exactly the liquid cell configuration where medium one and medium three are sapphire and medium two is air, carbon disulfide, nitromethane, hexane, etc.. In general, as the light passes through any interface part of the beam is reflected and the rest is transmitted:

$$R + T = 1 \quad \text{B.1}$$

where R is the reflectance and T is the transmittance. In terms of intensities,

$$I_2 = T_1 I_1 \quad \text{B.2}$$

When the angle of incidence is 0° , then the reflectance may be written as²¹:

$$R = \left[\frac{n_2 - n_1}{n_2 + n_1} \right]^2 \quad \text{B.3}$$

and the transmittance is²¹:

$$T = \frac{4n_1 n_2}{(n_2 + n_1)^2} \quad \text{B.4}$$

Here, n_1 and n_2 are the index of refraction of mediums 1 and 2. When the ray passes through the second interface into medium number three, again part of the ray is reflected and the rest transmitted and

$$I_3 = T_2 I_2 \quad \text{B.5}$$

I_3 is the intensity that is actually measured in our experiments. I_1 is the beam intensity prior to any reflectance/transmittance from any interface. It is possible to relate I_3 to I_1 by

$$I_3 = T_1 T_2 I_1 \quad \text{B.6}$$

or

$$I_1 = \frac{1}{T_1} \frac{1}{T_2} I_3 \quad \text{B.7}$$

By calculating $\frac{1}{T_1}$, $\frac{1}{T_2}$, measuring I_3 and then multiplying all three together, I_1 is determined. I_1 is the light intensity transmitted through the liquid cell but corrected for reflection losses. To calculate an absorption spectra, this correction technique must be applied to both the reference spectra and to the sample spectra.

As stated above, the refractive index of the materials is needed. The refractive index of sapphire is given by²:

$$n^2 - 1 = \sum_{m=1}^3 \frac{A_m \lambda^2}{\lambda^2 - \lambda_m^2} \quad \text{B.8}$$

where

A_1	=	1.023798	λ_1^2	=	0.00377588
A_2	=	1.058264	λ_2^2	=	0.0122544
A_3	=	5.280792	λ_3^2	=	321.3616

and the wavelength is in microns. Table B.1 lists index values for hexane, and nitromethane²². These values were fit to a straight line with the fitting parameters also listed.

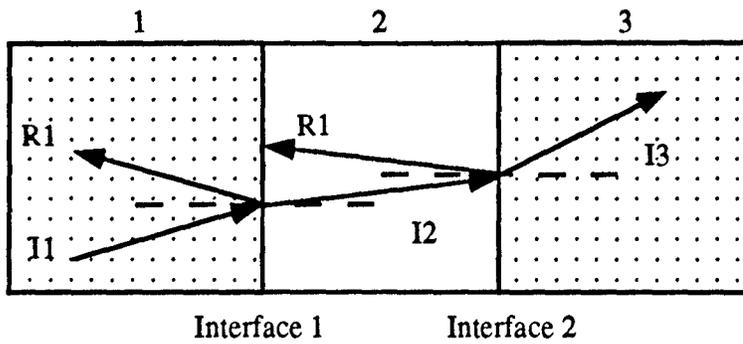
Figure B.2 shows the corrected transmission spectra for hexane and air (both spectra have been corrected). Since neither air nor hexane absorb photons in this wavelength region, the corrected spectra should be identical. The comparison is quite good except at the peak intensity. The most likely cause of this deviation is the pulse-to-pulse fluctuation of the flashlamp.

This same procedure may be applied to any other liquid sample (i.e., carbon disulfide and nitromethane) as long as the wavelength dependency of the refractive index is known. Based upon the hexane data it seems reasonable to continue and apply this correction.

Table B.1: Refraction Indices

Material	Wavelength (Angstroms)	n
Hexane	6563	1.37337
	5893	1.37536
	4861.5	1.37988
	4340.6	1.38365
Nitromethane	6563	1.37884
	5893	1.38133
	4861.5	1.38771
	4340.6	1.39305

Material	Slope	Intercept (angstroms)	R ²
Hexane	$-4.551364 \times 10^{-06}$	1.402708	0.988
Nitromethane	$-6.309748 \times 10^{-06}$	1.419397	0.985



$$I_2 = T_1 * I_1$$

$$I_3 = T_2 * I_2$$

$$I_3 = T_1 * T_2 * I_1$$

$$I_1 = \frac{1}{T_1} * \frac{1}{T_2} * I_3$$

Figure B.1: Schematic reflection losses at the interfaces. I_1 is the incident beam, I_2 is the transmitted beam, I_3 is the experimentally measured beam, R_1 is the portion of I_1 reflected from interface #1, and R_2 is the portion of I_2 reflected from interface #2.

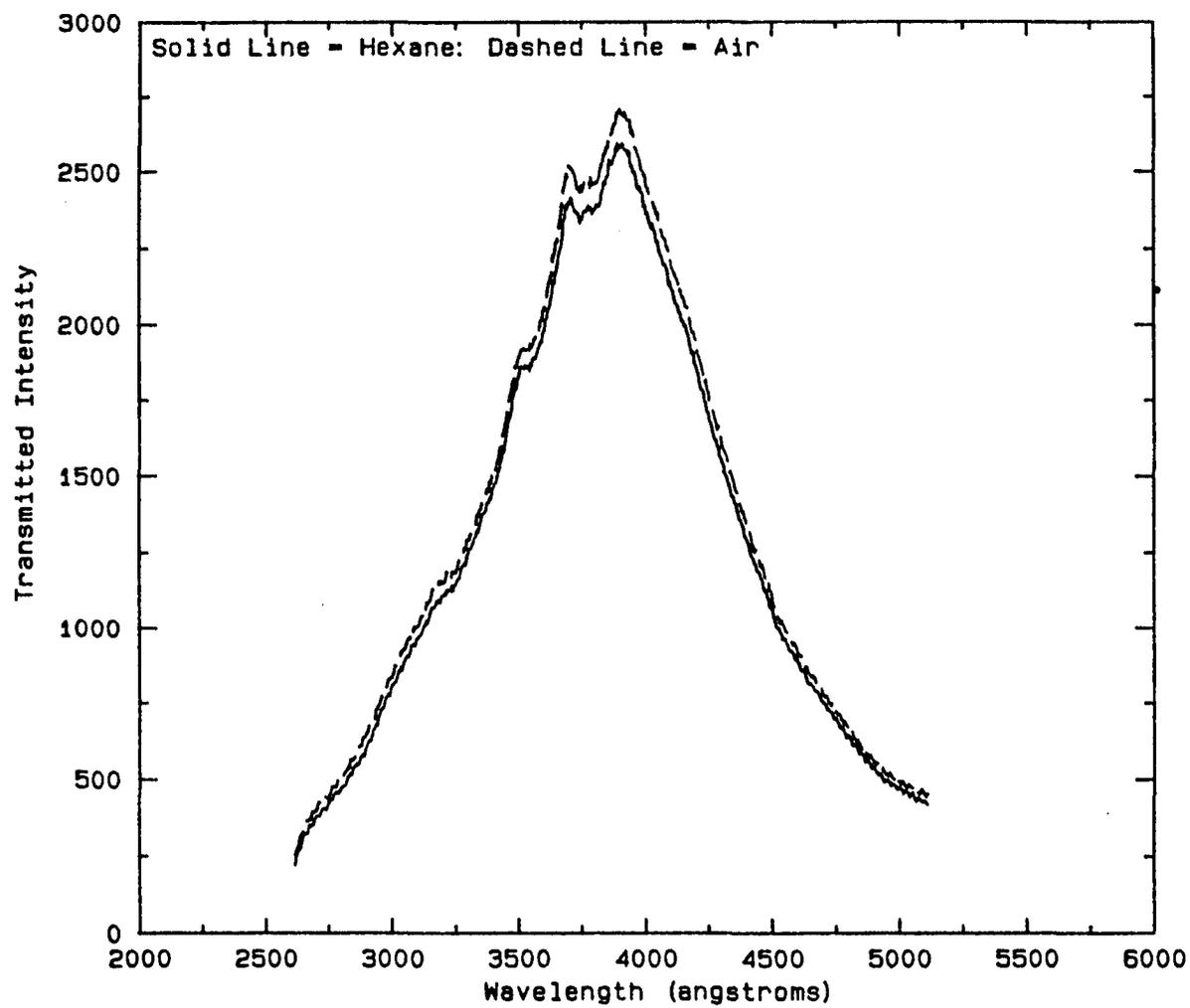


Figure B.2: Reflection corrected hexane transmission spectra.

Appendix C: Data Analysis

C.1 Introduction

A Hewlett-Packard 9000 computer is used for the data analysis of the transmission spectra. New data analysis programs have been written and used. These programs are included at the end of this appendix.

The main program (ABSFIT) has many options. Briefly, it takes the binary OMA data and converts it first to ASCII data and then to either track or channel data. Included is a subroutine to apply the reflection corrections described earlier. This program is intended for liquid cells containing air, hexane, or nitromethane. The program also calculates absorbances based upon reference data from another data file or from within its own file (important for shot data). Also, wavelength shifts in the absorption data may be calculated. The program can smooth, graph, and plot the data. Before describing specific details of the program, it is necessary to show what the program is doing. This involves a short discussion of the absorption process. (Note: the source codes for the programs of ABSFIT are located in the directory */users/kelly/fortran/prefit*. The executable codes are located in */users/kelly/bin*.)

C.2 Beer's Law

When monochromatic radiation passes through an absorbing species, part of the power (or energy) is absorbed and the remainder is transmitted. If P is the transmitted power and P_0 is the incident power, then the ratio of P/P_0 is the transmittance of the sample:

$$P = T * P_0 \quad \text{C.1}$$

If $T = 1$ the species is perfectly transparent and no absorption occurs. If $T = 0$ the species is a perfectly absorbing system and no radiation is transmitted. The loss in power depends upon the concentration of absorbing species as well as the path length the beam travels. The mathematical expression of these concepts is Beer's law²³⁻²⁵:

$$\log\left(\frac{P_0}{P}\right) = \epsilon * b * c = A \quad \text{C.2}$$

where

- c = concentration of absorbing species, usually expressed as moles per liter.
- b = thickness of sample that the beam passes through, usually expressed in cm.
- ϵ = molar absorptivity, usually expressed in liter per (mole - cm). ϵ is a proportionality constant characteristic of the absorbing material at the wavelength of measurement. It is also known as the molar extinction coefficient.
- A = absorbance of the material for a particular concentration, sample thickness, and wavelength. It is unitless and also known as optical density D or extinction E.

Absorbance is a measure of the energy absorbed by the sample. For a completely transparent system $A = 0$ and for the completely absorbing system $A = \text{infinity}$. Beer's law states that the absorbance of a substance is directly proportional to the concentration and to the cell thickness. There are limitations however which need to be considered²⁴⁻²⁹.

- (1) Beer's law of absorption is successful only for dilute solutions (usually < 0.01 molar). At higher concentrations concentration-dependent interactions are possible, i.e., aggregation of dyes, polymerization, etc..
- (2) The absorbing species may react with the solvent, impurities in the solution, or the container. If a reaction occurs or if the absorbing species dissociates, then the actual concentration of the solution may be more or less than that calculated.
- (3) Noncollimated light incident upon the absorption cell can cause problems. Beer's law was derived on the basis that all rays of light travel through the same cell thickness. For nonparallel light some rays will travel longer distances than other rays. More intensity will be subtracted from these rays and the calculated absorbance will be greater than the true absorbance.
- (4) Use of non-monochromatic light can cause deviations from Beer's law. The molar absorptivity of a substance is different for different wavelengths. However, it has been shown that use of polychromatic radiation results in only small deviations from Beer's law provided the absorbing species does not show large changes in the molar absorptivity as a function of wavelength. In other words, the calculated molar absorptivity of a band edge when using polychromatic light is suspect. The calculated molar absorptivity of the peak absorbance is probably accurate even when using polychromatic light.

These limitations mean that when measuring the absorbance of a pure (highly concentrated) solution care has to be taken when trying to calculate molar absorptivities. Also, since a

polychromatic light source is used, molar absorptivities of the absorbance peaks in dilute solutions may be calculated. It is possible to improve the accuracy of absorption measurements²⁶⁻²⁹. The minimum error inherent in Beer's law is calculated below.

The relative error in Beer's Law ($\Delta A/A$) may be evaluated as follows:

$$\log\left(\frac{I}{I_0}\right) = -\epsilon * b * c = -A \quad \text{C.3}$$

Differentiating,

$$d\left(\log\left(\frac{I}{I_0}\right)\right) = d(-A) \quad \text{C.4}$$

giving

$$\frac{\log(e) * d\left(\frac{I}{I_0}\right)}{\frac{I}{I_0}} = -dA. \quad \text{C.5}$$

Dividing by -A,

$$\frac{-dA}{-A} = \frac{\log(e) * d\left(\frac{I}{I_0}\right)}{(-A) * \left(\frac{I}{I_0}\right)} = \frac{-\log(e) * d\left(\frac{I}{I_0}\right)}{\left(\log\frac{I}{I_0}\right) * \left(\frac{I}{I_0}\right)} \quad \text{C.6}$$

The value of the error function may be approximated by substituting finite values for the infinitesimals:

$$\frac{\Delta A}{A} = \frac{-\log(e) * \Delta \left(\frac{I}{I_0} \right)}{\left(\log \frac{I}{I_0} \right) * \left(\frac{I}{I_0} \right)} \quad \text{C.7}$$

$\Delta(I/I_0)$ is a constant value. Differentiating C.7 with respect to I/I_0 and setting the function equal to zero gives the point of minimum error:

$$\frac{d \left(\frac{\log(e) * \Delta \left(\frac{I}{I_0} \right)}{\left(\log \frac{I}{I_0} \right) * \left(\frac{I}{I_0} \right)} \right)}{d \frac{I}{I_0}} = \frac{-0.4343 \Delta \left(\frac{I}{I_0} \right) \left(0.4343 + \log \left(\frac{I}{I_0} \right) \right)}{\left(\frac{I}{I_0} \log \left(\frac{I}{I_0} \right) \right)^2} \quad \text{C.8}$$

or,

$$-\log \left(\frac{I}{I_0} \right) = 0.4343 \quad \text{C.9}$$

and finally

$$\frac{I}{I_0} = 0.368. \quad \text{C.10}$$

The minimum absorbance error is achieved when solutions of 36.8% transmittance are used²⁶⁻²⁹. However, a plot of relative error as a function of transmittance (see Figure C.1) shows in the 20-65% transmittance region, there is a nearly constant error.

C.3 Computer Programs

Table C.1 lists the different routines used in *ABSFIT*. Some of the programs are written in the 'C' programming language with in the rest in 'FORTRAN'. The C programs are used mainly for system calls.

C.3.1 ABSFIT Programs

The following routines are used in the program *ABSFIT*.

(1) *Ambient*: This routine calculates the absorbance of a sample data file. The sample is referenced to another file.

```
Subroutine Ambient
C
C
DIMENSION TR(550),A(550)
DIMENSION REF(550),BKG(550),CHAN(550)
CHARACTER NFILE*64
C
C
50 Call Clear
Type 51
51 Format(//,20X,'SINGLE TRACK AMBIENT DATA ANALYSIS',//)
C
190 TYPE 200
200 FORMAT($,///,'Input name of AMBIENT data file: '
*' (? for Directory, Q to Quit): ')
READ(5,1200,ERR=50)NFILE
If (NFILE.EQ.'?') THEN
Call Directory
ELSEIF (NFILE.EQ.'Q') THEN
RETURN
ENDIF
IF (NFILE.EQ.'?') GOTO 50
OPEN(UNIT=9,FILE=NFILE,STATUS='OLD',ERR=9999)
DO I=1,500
READ(9,2000)CHAN(I),TR(I)
ENDDO
CLOSE(UNIT=9)
C
C
390 TYPE 391
391 FORMAT($,'Input name of REF data file: '
*' (? for Directory): ')
READ(5,1200,ERR=390)NFILE
```

```

      If (NFILE.EQ.'?') THEN
        Call Directory
      ENDIF
      IF (NFILE.EQ.'?') GOTO 390
      OPEN(UNIT=9,FILE=NFILE,STATUS='OLD',ERR=9999)
      DO I=1,500
        READ(9,2000)CHAN(I),REF(I)
      ENDDO
      CLOSE(UNIT=9)
C
C
420  TYPE 430
430  FORMAT($,'Input name of BKG data file: '
      *' (? for Directory): ')
      READ(5,1200,ERR=420)NFILE
      If (NFILE.EQ.'?') THEN
        Call Directory
      ENDIF
      IF (NFILE.EQ.'?') GOTO 420
      OPEN(UNIT=9,FILE=NFILE,STATUS='OLD',ERR=9999)
      DO I=1,500
        READ(9,2000)CHAN(I),BKG(I)
      ENDDO
      CLOSE(UNIT=9)
C
C
      Calculate Absorbance
C
      DO J=1,500
        T0 = REF(J) - BKG(J)
        IF (T0.LE.0.0) THEN
          T0 = 1.0
        ENDIF
        T = TR(J) - BKG(J)
        IF (T.LE.0.0) THEN
          T = 1.0
        ENDIF
D      print*,'J,Ref,Bkg = ',J,REF(J),BKG(J)
D      print*,'Tr,Bkg = ',TR(J),BKG(J)
        TT = T0/T
D      PRINT*,'T0,T,T0/T = ',T0,T,TT
        A(J) = ALOG10(T0/T)
D      PRINT*,'I,Abs = ',J,A(J)
D      print*,'*****'
      ENDDO
C
      XMAX = -10000.
      XMIN = 10000.
      DO I=1,500
        IF (CHAN(I).LT.XMIN) XMIN = CHAN(I)
        IF (CHAN(I).GT.XMAX) XMAX = CHAN(I)
      ENDDO
C
      YMAX = -10000.

```

```

        YMIN = 10000.
        DO i=1,500
            IF (A(I).LT.YMIN) YMIN = A(I)
            IF (A(I).GT.YMAX) YMAX = A(I)
        ENDDO
C
        XINC = (XMAX-XMIN)/5.
        XNUM = XINC/2.
        YINC = (YMAX-YMIN)/5.
        YNUM = YINC/2.
        CALL GRAPH1(CHAN,A,500,'Wavelength (angstroms)',
        *'Absorbance','Absorption Data',XMIN,XMAX,
        * XNUM,XINC,YMIN,YMAX,YNUM,YINC,2,0,1)
C
520     TYPE 530
530     FORMAT($,'Input name of ABS data storage file: (Q to'
        *' Quit) ')
        READ(5,1200,ERR=520)NFILE
        IF (NFILE.EQ.'Q') THEN
            GOTO 9998
        ENDIF
        OPEN(UNIT=9,FILE=NFILE,STATUS='NEW',ERR=9999)
        DO I=1,500
            WRITE(9,2000)CHAN(I),A(I)
        ENDDO
        CLOSE(UNIT=9)
C
        GOTO 9998
C
1200    FORMAT(A)
2000    FORMAT(1p,2e15.6)
9999    Print*,'Error has occurred...'
9998    Return
        END

```

(2) *Channel*: Converts the ASCII data into intensity versus tracks (or time), while the channel (or wavelength) is held constant.

```

Subroutine Channel
*
*           VARIABLE DEFINITIONS           *
*
integer ntrack,nchan,vchan,i,j
character*20 filein, fileout
real*4 fdat(501,501)
character*1 ANSWER
10      data ntrack,nchan /500,500/
*
*           GET INPUT DATA           *
20      Call Clear

```

```

Type 32
32  FORMAT(//,20X,'Channel Subroutine',//)
    write(6,1070)
Type 5000
5000 Format($,/, 'Enter name of input file: (? for directory '
      *' Q to Quit) ')
      read(5,1000,err=21,end=2000)filein
      IF (filein.eq.?) then
        Call Directory
      ELSEIF (FILEIN.EQ.'Q') Then
        Return
      Endif
      IF (filein.eq.?) then
        goto 20
      endif
      open(unit=10,file=filein,status='old',err=22)
      goto 30

21  write(6,1010)
     goto 20

22  write(6,1020)
     goto 20

30  write(6,1090)filein
     do 40 i=1,501
       read(10,*,err=45,end=45) (fdat(i,j),j=1,500)

40  continue
     write(6,1070)
Type 5100
5100 Format($,/'File does not comply with format. Please'
      *' re-enter')
     close(10)
     goto 20

45  ntrack=i-1

100 write(6,1070)
Type 5700
5700 Format($,/'Enter name of output file: ')
     read(5,1000,err=101,end=2000)fileout
     open(unit=2,file=fileout,status='new',err=102)
     goto 300

101 write(6,1010)
     goto 100

102 write(6,1020)
     goto 100

```

```

300  write(6,1060)nchan
      read(5,*,end=2000,err=301)vchan
      if ((vchan .le. (nchan)) .and. (vchan .ge. 1))
        *then
          goto 310
        endif

301  write(6,1010)
      goto 300

310  do 320 i=1,ntrack
          write(2,1050) i,fdat(i,vchan)
320  continue
      close(2)
      write(6,1080)fileout

5900  Type 6000
6000  Format($,/, 'Create another channel file? Y or N: ')
      Read(5,800)ANSWER
      IF (ANSWER.EQ.'Y') GOTO 100
      IF (ANSWER.EQ.'y') GOTO 100

      Goto 2000

800  Format(A1)
900  format(a2)
1000 format(a20)
1010 format(/'Invalid entry, please re-enter')
1020 format(/'Unable to open named file, please re-enter')
1060 format($,/'Enter channel number (1 - ',i3,'). ')
1050 format(1p,2e15.6)
1070 format(/)
1080 format(/,a40,'created')
1090 format(/'LOADING ',a20)

2000 close(10)
      close(2)
      Return
      End

```

(3) *Clear*: Sends a system command, clearing the terminal's screen.

```

#include <stdio.h>

clear()
{
    system("clear");
    return(0);
}

```

(4) *Correction*: Makes reflection corrections to the data of a single file. If corrections are needed for multiple files, calls Multicorrect to do so.

```

Subroutine Correction
C
C
      IMPLICIT DOUBLE PRECISION (D)
      DIMENSION OLDDATA(550),NEWDATA(550)
      DIMENSION CHAN(550),DSAPPHIRE(550)
      DIMENSION DINDEX(3,550)
      CHARACTER NFILE*64,Answer*1
      Integer Choice
C
C
90    Call Clear
      Type 91
91    FORMAT(//,20X,'Reflection Correction',//)
      Type 100
100   Format($,////,'Do you want to:',//,
        *20X,'(1) Correct a single file',//,
        *20X,'(2) Correct multiple files.',//,
        *20X,'(3) Quit.',//,
        *20X,'Please input 1, 2, or 3: ')
      Read(5,1200)Answer
      If ((Answer.NE.'1').AND.(Answer.NE.'2').AND.
        *(ANSWER.NE.'3')) then
          goto 90
      ELSEIF (Answer.EQ.'1') then
          goto 190
      ELSEIF (Answer.EQ.'2') then
          goto 3000
      ELSEIF (Answer.EQ.'3') then
          Return
      Endif
C
190   TYPE 200
200   FORMAT($,////,'Input name of ASCII data file: (? for
        *' directory, Q to Quit) ')
      READ(5,1200,ERR=190)NFILE
      If (NFILE.EQ.'?') Then
          Call Directory
      ELSEIF (NFILE.EQ.'Q') then
          Return
      ENDIF
      IF (NFILE.EQ.'?') THEN
          Goto 190
      ENDIF
      OPEN(UNIT=9,FILE=NFILE,STATUS='OLD')
      DO I=1,500
          READ(9,2000)CHAN(I),OLDDATA(I)
      ENDDO
      CLOSE(UNIT=9)

```

```

C
C
300 Type 310
310 Format($,////,'Is the sample interface:',//,
      *20X,'(1) Sapphire - Air',//,
      *20X,'(2) Sapphire - Hexane',//,
      *20X,'(3) Sapphire - Nitromethane',//)
      Type 320
320 Format($,/,20X,'Please enter choice (1-3): ')
      Read(5,*)Choice
      If ((Choice.LT.1).OR.(Choice.GT.3)) GOTO 300

C
C
      Calculate index of refraction for sapphire
C
      A1 = 1.023798
      A2 = 1.058264
      A3 = 5.280792
      DA1 = 0.00377588
      DA2 = 0.0122544
      DA3 = 321.3616
      DO I = 1, 500
          CH = (CHAN(I)/10000)**2
          DNS1 = (A1*CH)/(CH-DA1)
          DNS2 = (A2*CH)/(CH-DA2)
          DNS3 = (A3*CH)/(CH-DA3)
          DSAPPHIRE(I) = SQRT(DNS1+DNS2+DNS3+1)
      Enddo

C
C
      If (Choice.EQ.1) then
C
      Calculate index of refraction for air
      DO I=1,500
          DNA1 = 12.288/(CHAN(I)*CHAN(I)*1E-08)
          DNA2 = 0.3555/((CHAN(I)**4)*1E-16)
          DNA3 = 2726.43
          DINDEX(1,I) = ((DNA1+DNA2+DNA3)*1E-07)+1
D
      PRINT*,I,CHAN,N = ',I,CHAN(I),DNA(I)
      ENDDO

C
      ELSEIF (Choice.EQ.2) THEN
C
      calculate index for hexane
      DO I=1,500
          DNH0 = 1.402708
          DNH1 = -4.551364E-06 * CHAN(I)
          DINDEX(2,I) = DNH0+DNH1
D
      PRINT*,I,Chan,N = ',I,CHAN(I),DNH(I)
      ENDDO

C
      ELSEIF (Choice.EQ.3) THEN
C
      calculate index of refraction for nitromethane
      DO I=1,500
          DNA0 = 1.419397
          DNA1 = -6.309748E-06 * CHAN(I)
          DINDEX(3,I) = DNA0 + DNA1

```

```

D      PRINT*,I,CHAN,N = ',I,CHAN(I),DNA(I)
      ENDDO
C
      ENDIF
C
C      Now correct spectra
C
      DO I=1,500
          TOP = (DINDEX(Choice,I)+DSAPPHIRE(I))**2
          BOT = 4*DINDEX(Choice,I)*DSAPPHIRE(I)
          NEWDATA(I) = OLDDATA(I) * ((TOP/BOT)**2)
      ENDDO
C
C
C
5520   TYPE 5530
5530   FORMAT($,///,'Input name of CORRECTED data storage file:')
      READ(5,1200,ERR=5520)NFILE
      OPEN(UNIT=9,FILE=NFILE,STATUS='NEW',ERR=9999)
      DO I=1,500
          WRITE(9,2000)CHAN(I),NEWDATA(I)
      ENDDO
      CLOSE(UNIT=9)
C
C
      Goto 9999
C
3000   Call Multicorrect

1200   FORMAT(A)
2000   FORMAT(1p,2e15.6)
9999   Return
      END

```

(5) *Directory*: Makes a system call, listing the contents of the current directory.

```

#include <stdio.h>

directory()
{
    char cmd[161], ch;

    while( cmd != "" )
    {
        system("ls");
        {
            system( cmd );
            puts("\n hit return to continue");
            scanf("%c", &ch);
        }
        return(0);
    }
}

```

```
}
```

(6) *Graph*: Makes a system call to IGRAPH.

```
#include <stdio.h>

graph()
{
    char igrph[169], fline[161];
    char *cmd = "igrph ";

    strcat( igrph, cmd );

    system("clear");

    puts("\nGRAPHING OPERATION\n");

    puts("Current directory:");
    system("ls");

    puts("\n Enter the files to submit to igrph:\n");
    printf(" ---> ");
    if( gets( fline ) == NULL )
    {
        puts("\n\n Invalid file name entries");
        system("clear");
        return(0);
    }
    else
        system( strcat( igrph, fline ) );

    system("clear");

    puts("\nGRAPHING OPERATION\n");
    puts("\n Plotting complete, returning to PREFIT.");
}
```

(7) *HELP*: The help screen.

Subroutine Help

```
1000 Call Clear
3000 Type 4000
5000 Type 6000
6000 Format($,/, 'This program is for calculating ABSORPTION '
      *'spectra. The raw data is taken as',/, 'shot or ambient data'
      *' using the Cordin system. ',/, 'Once the OMA data has '
      *'transferred from the OMA to the HP-9000 computer via '
      *'the ',/, 'TRANSFER program, this program converts ')
```

```

*the OMA, binary data into ASCII data. 'This data is '
*then converted into either track versus wavelength or '
*channel '(wavelength) versus time (track) data. ',
*//, 'The sequence of steps to use this program is '
*typically: '//,5x, 'Option 1 - Convert the data',
*/,5x, 'Option 2 - Put the data into Tracks',
*/,5x, 'Option 4 - Correct the Track data for Reflections',
*/,5x, 'Option 5 - Calculate the absorption spectra.',
*//, 'Option 3 (Channel data) is used to confirm the '
*'Impact track. 'Option 6 (Delta Absorption) is used'
*' to see changes during the shot.', 'Option 7 - '
*'Absorption band shift will calculate the shift of '
*'the band edge.', 'The other options are '
*'self-explanatory.', 'Hit RETURN to return to the main'
*' menu screen...')
Read(5,1200)ZZZ
1200 Format(A)
      Return
4000 Format(/, 'Help for the ABSFIT Program')
9999 STOP
      END

```

(8) *LINFIT*: The linear regression routine.

```

      Subroutine LinFit
C
C
      DIMENSION TR(550), CHAN(550), X(550)
      DIMENSION Y(550), XX(550)
      DIMENSION BaseFit(550), EdgeFit(550)
      CHARACTER NFILE*64, Title*64, IOption*1
C
C
50      Call Clear
      Type 51
51      Format(//, 20X, 'Shift Data Analysis - Single Track', //)
C
      Type 52
52      Format(///, 'This program will calculate the wavelength '
*'of the intersection of the linear extrapolated line of the'
*' absorption band edge with the'
*' base line.', //, 'The user is shown the absorption curve'
*', ' and is asked to input starting & stopping channels '
*'for the band edge and the base line.', //, 'This program'
*' is used for calculating band edge shifts.')
190      TYPE 200
200      FORMAT($, ///, 'Input name of ABSORPTION data file: '
*' (? for Directory, Q to Quit): ')
      READ(5, 1200, ERR=50) NFILE
      If (NFILE.EQ.'?') THEN
          Call Directory
      ELSEIF (NFILE.EQ.'Q') THEN
          GOTO 9998

```

```

ENDIF
IF (NFILE.EQ.'?') GOTO 50
OPEN(UNIT=9,FILE=NFILE,STATUS='OLD',ERR=9999)
DO I=1,500
    READ(9,2000)XX(I),TR(I)
ENDDO
CLOSE(UNIT=9)
C
    Flag = 1.
    DO I=1,500
        CHAN(I) = I
    ENDDO
C
1600 TYPE 1610
1610 FORMAT(//,'You are about to see a plot of your '
* 'data.',//,
*'You will then input start & stop channels... ')
Type 1622
1622 Format(//,'This is for the EDGE data...')
TYPE 1620
1620 FORMAT(//,'Hit RETURN when finished viewing ...')
Type 1621
1621 Format(//,'Hit RETURN to begin...')
Read(5,1200)ZZZ
C
1623 XMAX = -10000.
XMIN = 10000.
DO I=1,500
    IF (CHAN(I).LT.XMIN) XMIN = CHAN(I)
    IF (CHAN(I).GT.XMAX) XMAX = CHAN(I)
ENDDO
C
YMAX = -10000.
YMIN = 10000.
DO i=1,500
    IF (TR(I).LT.YMIN) YMIN = TR(I)
    IF (TR(I).GT.YMAX) YMAX = TR(I)
ENDDO
C
XINC = (XMAX-XMIN)/5.
XNUM = XINC/2.
YINC = (YMAX-YMIN)/5.
YNUM = YINC/2.
If (flag.eq.1.) then
    Title = 'Edge Data'
Elseif (flag.eq.2.) then
    Title = 'Base Data'
Endif
Y2=YMAX
Y3=YINC
Y4=YNUM
1625 CALL PLOT1(CHAN,TR,500,'Channel #',
*'Absorbance',Title,XMIN,XMAX,
* XNUM,XINC,YMIN,YMAX,YNUM,YINC,2,0,1)

```

```

1630 TYPE 1640
1640 FORMAT(//,'To see again type 1, ',/,
*'Type 2 to Continue, ',/, 'Type 3 to expand plot.')
TYPE 1641
1641 FORMAT($,/, 'Please input 1, 2, or 3: ')
XMin = 0.
READ(5,1200,ERR=1630)ANSWER
IF ((ANSWER.EQ.'1').AND.(XMin.EQ.0.)) THEN
GOTO 1625
ELSEIF((ANSWER.EQ.'1').AND.(XMin.NE.0.)) THEN
GOTO 899
ELSEIF (ANSWER.EQ.'2') THEN
GOTO 401
ENDIF
Type 900
900 Format($,/, 'Input Min Channel: ')
Read(5,905)XMin
Type 901
901 Format($,/, 'Input Max Channel: ')
Read(5,905)XMax
Type 902
902 Format($,/, 'Input Min Intensity: ')
Read(5,905)YMin
Type 903
903 Format($,/, 'Input Max Intensity: ')
Read(5,905)YMax
905 Format(F10.2)
Xinc = (XMax-XMin)/5.
Xnum = Xinc/2.
Yinc = (Ymax-Ymin)/5.
Ynum = Yinc/2.
899 CALL PLOT1(CHAN,TR,500,'Channel #',
*'Absorbance',Title,XMIN,XMAX,
* XNUM,XINC,YMIN,YMAX, YNUM,YINC,2,0,1)

C
Goto 1630

C
401 Type 951
951 Format($,/, "Start at which channel? (i.e., 50.) ")
READ(5,1300)ChanStart
IF ((ChanStart.Lt.1.).OR.(ChanStart.GT.500.)) Then
Goto 401
Endif

400 Type 950
950 Format($,/, 'Stop at which channel? (i.e., 100.) ')
READ(5,1300)ChanStop
IF ((ChanStop.Lt.1.).OR.(ChanStop.GT.500.)) Then
Goto 400
Endif

C
M = ChanStop - ChanStart + 1

```

```

Do I=1,M
  Y(I) = TR(ChanStart + I - 1)
  X(I) = XX(ChanStart + I - 1)
ENDDO

Call Regress(X, Y, M, Z1, Z2)

IF (Flag.eq.1.) then
  Edge1 = Z1
  Edge2 = Z2
Elseif (Flag.eq.2.) Then
  Base1 = Z1
  Base2 = Z2
Endif
If (Flag.eq.1.) then
  Flag = 2.
  Type 1605
1605 Format (//, 'This was for the EDGE data. You '
*'will now do it for the BASE data.',// ' Hit '
*'RETURN to continue...')
  Read(5,1200)ZZZ
  Goto 1623
Endif
Wave = (Base2-Edge2)/(Edge1-Base1)
Write(6,1611)Wave
1611 Format (//,'The wavelength is ',F10.2,' angstroms.',//)
1612 Type 1612
1612 Format($,//,'Do you want to see the extrapolated fits?',//,
*20X,'Type a Y or N please: ')
READ(5,1200,Err=1613)IOption
IF ((IOPTION.NE.'Y').AND.(IOPTION.NE.'N')) THEN
  GOTO 1613
ELSEIF (IOPTION.EQ.'Y') THEN
Do I=1,500
  BaseFit(I) = Base1*XX(I) + Base2
  EdgeFit(I) = Edge1*XX(I) + Edge2
Enddo

XMAX = -10000.
XMIN = 10000.
DO I=1,500
  IF (XX(I).LT.XMIN) XMIN = XX(I)
  IF (XX(I).GT.XMAX) XMAX = XX(I)
ENDDO

C
XINC = (XMAX-XMIN)/5.
XNUM = XINC/2.
YMIN = 0.0
YMAX=Y2
YINC=Y3
YNUM=Y4

CALL PLOT1(XX,TR,500,'Wavelength ',
*'Intensity','Absorption Data',Xmin,Xmax,

```

```

* XNUM,XINC,YMIN,YMAX,YNUM,YINC,0,0,0)
CALL PLOT1(XX,BaseFit,500,'Wavelength ',
*'Intensity','Absorption Data',Xmin,Xmax,
* XNUM,XINC,YMIN,YMAX,YNUM,YINC,0,0,0)
CALL PLOT1(XX,EdgeFit,500,'Wavelength ',
*'Intensity','Absorption Data',Xmin,Xmax,
* XNUM,XINC,YMIN,YMAX,YNUM,YINC,1,0,1)

ELSEIF (IOPTION.EQ.'N') THEN
    Goto 9998
Endif

C
C

GOTO 9998

C
1300 FORMAT(F10.2)
1200 FORMAT(A)
2000 FORMAT(1p,2e15.6)
2010 FORMAT(1p,3e15.6)
9999 Print*,'Error has ocurred...'
9998 Return
END

```

(6) *Menu*: The main routine.

```

Program MENU

Integer Ioption

1000 Call Clear

3000 Type 4000

Type 2000
Type 2100
READ (5,*,err=1000)IANSWER
GOTO (10,20,30,40,50,60,70,80,90,100,110,120,130),IANSWER
Print*,"
PRINT*,'Invalid choice. Try again please.'
GOTO 1000

10 Call Clear
Type 101
101 Format(//,20X,'Convert OMA Binary Data to ASCII'
*' Data',//)
Call Omaconvert
Goto 1000

20 Call Track
GOTO 1000

```

```

30      Call Channel
        GOTO 1000

40      Call Correction
        GOTO 1000

50      Call Clear
        Type 5000
5000    Format($,/,,'Analyze Ambient Absorption Data',
        * ///,'Do you want to :',/,
        *20X,'(1) Analyze 1 track',/,
        *20X,'(2) Analyze many tracks ?',/,
        *20X,'Type a 1 or 2 please: ')
        READ(5,*,Err=100)IOption
        IF ((IOPTION.NE.1).AND.(IOPTION.NE.2)) THEN
            GOTO 50
        ELSEIF (IOPTION.EQ.1) THEN
            CALL AMBIENT
        ELSEIF (IOPTION.EQ.2) THEN
            CALL MULTIAMBIENT
        ENDIF
        GOTO 1000

60      Call Shot
        GOTO 1000

70      Call LinFit
        GOTO 1000

80      Call Clear
72      Type 71

71      Format($,/,,'Analyze Ambient Absorption Data',
        * ///,'Do you want to :',/,
        *20X,'(1) Smooth 1 file',/,
        *20X,'(2) Smooth many files ?',/,
        *20X,'Type a 1 or 2 please: ')
        Read (5,*,Err=72)Ianswer
        IF ((Ianswer.NE.1).and.(Ianswer.ne.2)) then
            Goto 80
        Elseif (Ianswer.eq.1) then
            Call smooth
        Elseif (Ianswer.eq.2) then
            Call multismooth
        Endif
        GOTO 1000

90      Call Clear
81      Type 82
82      FORMAT($,///,20X,'Simplemath',/////,
        * 'Do you want to: ',/,
        * 20X,'(1) Add 2 files',/,

```

```

* 20X,'(2) Subtract 2 files',/,
* 20X,'(3) Multiply 2 files',/,
* 20X,'(4) Divide 2 files')
Type 83
83  FORMAT($,/, 'Please input your selection (1-4): ')
    Read(5,*,Err=80)IANSWER
    If ((IANSWER.LT.1).or.(IANSWER.GT.4)) GOTO 90
    K = IANSWER
    Call Simplemath(k)
    Goto 1000

100  Call Graph
    Goto 1000

110  Call Shell
    Goto 1000

120  Call Help
    Goto 1000

130  Type 2200
2200 Format(///,20X,'Exiting program. Goodbye...')
    Goto 9999

2000 FORMAT($,////, 'Do you want to: ',/,
* 15X,'(1) Convert OMA data to ASCII data',/,
* 15X,'(2) Convert ASCII data to Track data',/,
* 15X,'(3) Convert ASCII data to Channel data',/,
* 15X,'(4) Correct TRACK data for Reflection losses',/,
* 15X,'(5) Calculate Absorption Spectra from '
*'Ambient (Track) data',/,
* 15X,'(6) Calculate DELTA Absorption Spectra from '
*'SHOT (Track) data',/,
* 15X,'(7) Calculate ABSORPTION band-edge shift',/,
* 15X,'(8) Smooth a file',/,
* 15X,'(9) Simplemath on 2 files',/,
* 14X,'(10) Igraph a file',/,
* 14X,'(11) Do a System Command',/,
* 14X,'(12) HELP with this program.',/,
* 14X,'(13) End Program.')
2100 FORMAT($,/, 'Please input your selection (1-13): ')
4000 Format(///, 'ABSFIT Program: For fitting Absorption'
*' data')
9999 STOP
    END

```

(10) *Multiambient*: This routine calculates the absorbance of a many, sample data files. The samples are referenced to other files.

Subroutine Multiambient

C

```

C
DIMENSION DATA(3,550),CHAN(550),ABS(550)
CHARACTER*64 NFILE(3),Outfile
CHARACTER*2 Prefix(4),Num,Answer
Integer Start,End,Count,Hi,Lo

C
50 Call Clear
TYPE 100
100 FORMAT(//,20X,'MUTI-TRACK AMBIENT ANALYSIS',//)
C
190 TYPE 200
200 FORMAT($,///,'Input name of AMBIENT data file prefix: '
*' (? for Directory, Q to Quit): ')
READ(5,1200,ERR=50)Prefix(1)
If (PREFIX(1).EQ.'?') THEN
    Call Directory
ELSEIF (PREFIX(1).EQ.'Q') THEN
    RETURN
ENDIF
IF (PREFIX(1).EQ.'?') GOTO 50

C
C
390 TYPE 391
391 FORMAT($,'Input name of REF data file prefix: '
*' (? for Directory): ')
READ(5,1200,ERR=390)Prefix(2)
If (PREFIX(2).EQ.'?') THEN
    Call Directory
ENDIF
IF (PREFIX(2).EQ.'?') GOTO 390

C
C
420 TYPE 430
430 FORMAT($,'Input name of BKG data file prefix: '
*' (? for Directory): ')
READ(5,1200,ERR=420)Prefix(3)
If (PREFIX(3).EQ.'?') THEN
    Call Directory
ENDIF
IF (PREFIX(3).EQ.'?') GOTO 420

C
820 TYPE 830
830 FORMAT($,'Input name of ABS data file prefix: '
*' (? for Directory): ')
READ(5,1200,ERR=820)Prefix(4)
If (PREFIX(4).EQ.'?') THEN
    Call Directory
ENDIF
IF (PREFIX(4).EQ.'?') GOTO 820

C
600 Type 610
610 Format($,///,'Input start track: ')
Read(5,1000,Err=600)Start
620 Type 630

```

```

630 Format($,/, 'Input end track: ')
    Read(5,1000,Err=620)End
    If(Start.GT.End) GOTO 600
    If ((Start.LT.1).OR.(End.GT.50)) Goto 600

C
900 Type 910
910 Format($,/, 'Have these data files been corrected for '
    *'Reflection losses? Y or N: ')
    Read(5,920,Err=900)ANSWER
920 FORMAT(A2)
    Do Count = Start,End
        Hi = int(Count/10)
        Lo = Count - Hi * 10
        Num = CHAR(Hi+48)//CHAR(Lo+48)
        Outfile = Prefix(4)//Num
        DO I=1,3
            IF (ANSWER.EQ.'Y') THEN
                Nfile(I) = Prefix(I)//Num//'.c'
            ELSE
                Nfile(I) = Prefix(I)//NUM
            ENDIF
            IF (I.EQ.3) then
                Nfile(I) = Prefix(I)//Num
            ENDIF
            OPEN(UNIT=9,FILE=NFILE(I),STATUS='OLD')
            DO J=1,500
                READ(9,2000)CHAN(J),DATA(I,J)
            ENDDO
        ENDDO

C
    Calculate Absorbance
C
    DO J=1,500
        T0 = DATA(2,J) - DATA(3,J)
        IF (T0.LE.0.0) THEN
            T0 = 1.0
        ENDIF
        T = DATA(1,J) - DATA(3,J)
        IF (T.LE.0.0) THEN
            T = 1.0
        ENDIF
        TT = T0/T
        ABS(J) = ALOG10(T0/T)
    ENDDO

C
    Store Data
C
    OPEN(UNIT=9,FILE=OUTFILE,STATUS='NEW',ERR=9999)
    DO I=1,500
        WRITE(9,2000)CHAN(I),ABS(I)
    ENDDO
    CLOSE(UNIT=9)
    Write(6,1300)OUTFILE

```

```

      ENDDO
C
      GOTO 9998
C
1000  FORMAT(I2)
1300  Format(//,20X,'New ',A6,' file created...')
1200  FORMAT(A)
2000  FORMAT(1p,2e15.6)
9999  Print*,'Error has occurred...'
9998  Return
      END

```

(11) *Multicorrect*: Makes reflection corrections to the data of a many files.

```

      Subroutine Multicorrect
C
C
      IMPLICIT DOUBLE PRECISION (D)
      DIMENSION OLDDATA(550),NEWDATA(550)
      DIMENSION CHAN(550),DSAPPHIRE(550)
      DIMENSION DINDEX(3,550)
      CHARACTER NFILE*64,Outfile*64
      CHARACTER*2 Prefix,Num,Extra
      Integer Choice,Start,End,Count,Hi,Lo
C
C
C
190   Call Clear
      Type 191
191   FORMAT(//,20X,'Multiple-Track Reflection Correction'//)
      TYPE 200
200   FORMAT($,////,'Input name of ASCII data file mask'
      *': (? for directory, Q to Quit) ')
      READ(5,1250,ERR=190)PREFIX
      If (PREFIX.EQ.'?') Then
          Call Directory
      ELSEIF (PREFIX.EQ.'Q') Then
          RETURN
      ENDIF
      IF (PREFIX.EQ.'?') THEN
          Goto 190
      ENDIF
C
400   Type 410
410   Format($,//,'Input start track: ')
      Read(5,1000,Err=400)Start
420   Type 430
430   Format($,//,'Input end track: ')
      Read(5,1000,Err=420)End
      If(Start.GT.End) GOTO 400

```

```

If ((Start.LT.1).OR.(End.GT.50)) Goto 400
C
300 Type 310
310 Format($,////,'Is the sample interface:',//,
*20X,'(1) Sapphire - Air',//,
*20X,'(2) Sapphire - Hexane',//,
*20X,'(3) Sapphire - Nitromethane',//)
Type 320
320 Format($/,20X,'Please enter choice (1-3): ')
Read(5,*)Choice
If ((Choice.LT.1).OR.(Choice.GT.3)) GOTO 300
C
Extra = '.c'
Do Count = Start,End
Hi = int(Count/10)
Lo = Count - Hi * 10
Num = CHAR(Hi+48)//CHAR(Lo+48)
Nfile = Prefix//Num
Outfile = Prefix//Num//Extra
OPEN(UNIT=9,FILE=NFILE,STATUS='OLD')
DO I=1,500
READ(9,2000)CHAN(I),OLDDATA(I)
ENDDO
C
C
C
C
Calculate index of refraction for sapphire
A1 = 1.023798
A2 = 1.058264
A3 = 5.280792
DA1 = 0.00377588
DA2 = 0.0122544
DA3 = 321.3616
DO I = 1, 500
CH = (CHAN(I)/10000)**2
DNS1 = (A1*CH)/(CH-DA1)
DNS2 = (A2*CH)/(CH-DA2)
DNS3 = (A3*CH)/(CH-DA3)
DSAPPHIRE(I) = SQRT(DNS1+DNS2+DNS3+1)
Enddo
C
C
If (Choice.EQ.1) then
C
Calculate index of refraction for air
DO I=1,500
DNA1 = 12.288/(CHAN(I)*CHAN(I)*1E-08)
DNA2 = 0.3555/((CHAN(I)**4)*1E-16)
DNA3 = 2726.43
DINDEX(1,I) = ((DNA1+DNA2+DNA3)*1E-07)+1
D
PRINT*,I,CHAN,N = ',I,CHAN(I),DNA(I)
ENDDO
C
ELSEIF (Choice.EQ.2) THEN
C
calculate index for hexane

```

```

DO I=1,500
    DNH0 = 1.402708
    DNH1 = -4.551364E-06 * CHAN(I)
    DINDEX(2,I) = DNH0+DNH1
D PRINT*,I,Chan,N = ',I,CHAN(I),DNH(I)
  ENDDO
C
  ELSEIF (Choice.EQ.3) THEN
C calculate index of refraction for nitromethane
  DO I=1,500
    DNA0 = 1.419397
    DNA1 = -6.309748E-06 * CHAN(I)
    DINDEX(3,I) = DNA0 + DNA1
D PRINT*,I,CHAN,N = ',I,CHAN(I),DNA(I)
  ENDDO
C
  ENDIF
C
C Now correct spectra
C
  DO I=1,500
    TOP = (DINDEX(Choice,I)+DSAPPHIRE(I))**2
    BOT = 4*DINDEX(Choice,I)*DSAPPHIRE(I)
    NEWDATA(I) = OLDDATA(I) * ((TOP/BOT)**2)
  ENDDO
C
C Store corrected data
C
9) OPEN(UNIT=9,FILE=OUTFILE,STATUS='NEW',ERR=999
    DO I=1,500
      WRITE(9,2000)CHAN(I),NEWDATA(I)
    ENDDO
    CLOSE(UNIT=9)
    Write(6,1300)Outfile
  ENDDO
C
C
C
1000 Format(I2)
1200 FORMAT(A)
1250 FORMAT(A2)
1300 Format(//,20X,'New ',A6,' file created...')
2000 FORMAT(1p,2e15.6)
9999 Return
END

```

(12) *Multismooth*: Smooth multiple files.

```

Subroutine Multismooth
C
C
REAL*4 OLDDATA(550),NEWDATA(550)
DIMENSION CHAN(550)
CHARACTER NFILE*64,Outfile*64
CHARACTER*2 Prefix,Num,Extra
Integer Start,End,Count,Hi,Lo

C
C
C
190 Call Clear
Type 191
191 FORMAT(//,20X,'Multiple-Track Smoothing',//)
TYPE 200
200 FORMAT($,////,'Input name of ASCII data file mask'
*': (? for directory, Q to Quit) ')
READ(5,1250,ERR=190)PREFIX
If (PREFIX.EQ.'?') Then
    Call Directory
ELSEIF (PREFIX.EQ.'Q') Then
    Return
ENDIF
IF (PREFIX.EQ.'?') THEN
    Goto 190
ENDIF

C
400 Type 410
410 Format($,/, 'Input start track: ')
Read(5,1000,Err=400)Start
420 Type 430
430 Format($,/, 'Input end track: ')
Read(5,1000,Err=420)End
If(Start.GT.End) GOTO 400
If ((Start.LT.1).OR.(End.GT.50)) Goto 400

C
write(6,1160)
read(5,*)k
k=int(k/2)

C
Extra = 's'
Do Count = Start,End
    Hi = int(Count/10)
    Lo = Count - Hi * 10
    Num = CHAR(Hi+48)//CHAR(Lo+48)
    Nfile = Prefix//Num
    Outfile = Prefix//Num//Extra
    OPEN(UNIT=9,FILE=NFILE,STATUS='OLD')
    DO I=1,500
        READ(9,2000)CHAN(I),OLDDATA(I)
    ENDDO

C

```

C
C

```
do 710 j=1,500
  NEWDATA(j)=0
  if (j.le.k) then
    l=1
    m=2*k-j
    goto 703
  else
    l=j-k
  endif

  if (500.le.(j+k)) then
    m=500
    l=2*j-500
    goto 703
  else
    m=j+k
  endif
703  do 705 i=l,m
    NEWDATA(j)=NEWDATA(j)+OLDDATA(i)

705  continue
    NEWDATA(j) = NEWDATA(j) / (m-l+1)
710  continue
```

C
C
C

Store corrected data

```
OPEN(UNIT=9,FILE=OUTFILE,STATUS='NEW',ERR=9999)
D Print*, 'k = ',k
DO I=1,500
D PRINT*, I,Olddata,Newdata=',I,OLDDATA(I),NEWDATA(I)
WRITE(9,2000)CHAN(I),NEWDATA(I)
ENDDO
CLOSE(UNIT=9)
Write(6,1300)Outfile

ENDDO
```

C
C
C

```
1160 format($,/'Enter smooth window length in points: ')

1000 Format(I2)
1200 FORMAT(A)
1250 FORMAT(A2)
1300 Format(//,20X,'New ',A6,' file created...')
2000 FORMAT(1p,2e15.6)
9999 Return
END
```

(13) *OmaConvert*: Converts the OMA's binary data into ASCII data.

```
#include <stdio.h>

omaconvert()
{
    char convert[80], arg1[20], arg2[20];
    char *cmd = "/users/kelly/bin/cvrt50 ";

    strcat(convert, cmd);

    printf("Enter filename of the binary (OMA) file to be converted:\n");
    while (gets(arg1) == NULL)
        printf("Binary filename:\n");
    strcat(convert, arg1);

    strcat(convert, " > ");

    printf("Enter the filename of the ASCII output file:\n");
    while (gets(arg2) == NULL)
        printf("ASCII filename:\n");
    strcat(convert, arg2);

    system(convert);
}
```

As seen, this routine calls an executable file named *CVRT50* is a C program with 3 subroutines: *get_b32_lbin.c*, *lbin_tbl.c*, and *omadat.h*. These files are listed below.

(13.1) *get_b32_lbin.c*

```
/* This subroutine is designed to read 32 bit integer data from raw */
/* OMA files. The routine will read four bytes into a union buffer and*/
/* export the buffer as a single integer.                               */

long get_b32_lbin(file_ptr)
FILE *file_ptr;
{
    CONVERT buf;
    int n_bytes=4, temp_c;

    while(n_bytes>0){
        if((temp_c=fgetc(file_ptr))!=EOF){
            buf.c[4-n_bytes--]=temp_c;
        }
        else{
            fprintf(stderr, "\nError get_b32_lbin\n");
            exit(1);
        }
    }
}
```

```

    }
return(buf.l);
}

```

(13.2) *lbin_tbl.c*

```

/*      This program is designed to read 32 bit integer      */
/*      numbers in an IEEE standard stored in binary format and */
/*      place the numbers in an ASCII format in a table directed */
/*      to standard output. This output can be redirected to a */
/*      file which may be read by other programs.      */

/*      Written by : Michael S Ives      */
/*      Date :      22 SEP 1987      */
/*      For :      Shock Dynamics Laboratory      */
/*      HP9000/350 Series under HP-UX UNIX Ver 5.2      */
/*      Modified :  25 SEP 1987 for Dr. Yoo      */

#include <stdio.h>
#include "omadat.h"
#include "get_b32_lbin.c"

main(argc,argv)
int argc;
char *argv[];
{
FILE *file1_ptr;
long int k,n;
long get_b32_lbin();

/* Open and check the specified files.      */

if(argc<2){
    fprintf(stderr,"\nError -- Incorrect number of arguments\n");
    fprintf(stderr,"Syntax is -- %s file1\n",argv[0]);
    exit(-1);
}
if((argv[1]==NULL)||((file1_ptr=fopen(argv[1],"r"))==NULL)){
    fprintf(stderr,"\nError -- Cannot open input file %s\n",argv[1]);
    exit(-1);
}

/* Write to the specified files the ASCII numbers      */

for(k=1;k<=50;k++){
    for(n=1;n<=500;n++){
        fprintf(stdout,"%ld ",get_b32_lbin(file1_ptr));
    }
    fprintf(stdout,"\n");
}

/* Perform final housekeeping by closing all open files.      */

```

```
fclose(file1_ptr);
}
```

(13.3) *omadat.h*

```
/* Header file for OMA data conversion routines. */
```

```
typedef union {
    char c[4];
    float f;
    long l;
} CONVERT;
```

(14) *Plot1*: A graphics routine for the CIT-414A terminals.

```
#include <stdio.h>

#define STRLEN 100
#define TRUE 1
#define FALSE 0
#define NONE 0L
#define FRAME 1L
#define GRID 2L
#define SCREEN 0L
#define PLOTTER 1L
#define not !
#define XSCALE 9.28
#define YSCALE 7.15

static int pipe_open = FALSE; /* pipe opened flag */
static FILE *pipe, *pcmd; /* pipe pointer */
FILE *popen();

plot1 (
    xarr, yarr, numpoints, /* arrays of x and y vals + num */
    xtitle, /* x axis title of graph */
    ytitle, /* y axis title */
    gtitle, /* graph title */
    xmin, xmax, /* min and max values for graph */
    xinc, xnum, /* x axis major and minor increments */
    ymin, ymax, /* min and max values for graph */
    yinc, ynum, /* y axis major and minor increments */
    frametype, /* type of frame for graph */
    dest, /* destination of graph */
    do_close, /* bool for overlays */
    xtlen, ytlen, gtlen /* lengths of strings */
)

float *xarr, *yarr;
char *xtitle, *ytitle, *gtitle;
int *numpoints;

float *xmin, *xmax, *ymin, *ymax;
```

```

float *xinc, *xnum, *yinc, *ynum;

int *frametype, *dest, *do_close;
long int xtlen, ytlen, gtlen;
{
    int i;
    int ominx, omaxx, odivx, ominy, omaxy, odivy, odivnx, odivny;
    char xt[STRLEN], yt[STRLEN], gt[STRLEN];
    char buf[1025], *cmd, *outfile;
    FILE *fin, *fopen();

    if (not pipe_open) {
        if (*dest == PLOTTER)
            cmd = "hpplot -p 1 | lpr -dhpplot";
        else
            cmd = "tek";

        pcmd = popen(cmd, "w");
        pipe_open = TRUE;
    }

    outfile = "/tmp/graphfile";

    normalize(xtitle, xtlen, xt); /* go fix those F77 strings */
    normalize(ytitle, ytlen, yt);
    normalize(gtitle, gtlen, gt);

    if (*dest == SCREEN)
        *yt=NULL;

    /* make up proper command */
    sprintf(buf, "graph -b -x %g %g %g -y %g %g %g -r .2\
-u .08 -n %g %g -w %g -h %g -s -lg '%s' '%s' '%s'\
-g %d > %s", *xmin, *xmax, *xinc, *ymin, *ymax, *yinc, *xnum, *ynum,
(*dest==SCREEN) ? (1.1) : (7.0/XSCALE),
(*dest==SCREEN) ? (.8) : (6.25/YSCALE),
xt, yt, gt, *frametype, outfile);

    pipe=popen(buf, "w");

    /* now print array contents to the command pipe */
    for (i=1; i<(*numpoints); i++) {
        fprintf(pipe, "%f %f\n", *(xarr+i), *(yarr+i));
    }
    pclose(pipe);

    fin = fopen(outfile, "r"); /* read the data in */
    while (!feof(fin)) /* print it to the pipe */
        fputc((fgetc(fin)), pcmd);
    fclose(fin);
    unlink(outfile);

    /* if close, send data to where it belongs */
    if(*do_close) {

```

```

        pclose(cmd);
        if(*dest == SCREEN) {
            getchar();
            cls();
        }
        pipe_open=FALSE;
    }
}

```

```

static normalize(fstring, len, cstring)
char fstring[STRLEN], cstring[STRLEN];
register unsigned long len;
{
    /*
     * This procedure normalizes the fortran strings to those compatible
     * with C (i.e. null terminated).
     */

    register int i;
    for(i=0; (i<len) && (i<STRLEN); i++)
        cstring[i] = fstring[i];
    cstring[i] = NULL;
    for (i=strlen(cstring)-1; i>0; --i) {
        if (cstring[i] == ' ')
            cstring[i] = NULL;
        else
            i= -1;
    }
}

static cls()
{
    system("clear");
}

```

(15) *Regress*: The linear regression routine.

```

subroutine regress(x,y,m,z1,z2)
real*4 x(10000), y(10000), a(42), c(6)

data maxnum /10000/

write(6,1000) m
1000 format(i5,' observations were read '/')
n = 1
2000 format(/' Fitting a polynomial of order',i3)

call stplrg (m, x, y, n, c, iflag, a)
if(iflag .eq. 1) then
do 200 i=1,n+1
k = n - i + 1

```

```

200    continue
        z1 = c(1)
        z2 = c(2)
        else
        go to 920
        endif

800    return
920    return
        end

```

(10) *Shell*: A system call, allowing the user to input any system command.

```

#include <stdio.h>

shell()
{
    char cmd[161], ch;

    while( cmd != "!" )
    {
        system("clear");
        puts("\nSHELL FOR EXECUTING SYSTEM COMMANDS");
        puts("(enter Z to return)\n");

        printf("\nCOMMAND: ");
        gets(cmd);

        if( cmd[0] != 'Z' )
        {
            system( cmd );
            puts("\n hit return to continue");
            scanf("%c", &ch);
        }
        else
        {
            system("clear");
            puts("\nSHELL FOR EXECUTING SYSTEM COMMANDS");
            puts("\n\n Closing Shell and returning to PREFIT.");
            return(0);
        }
    }
}

```

(17) *Shot*: This routine calculates the change in absorbance of sample data files. The samples are referenced to the same data, recorded at earlier times.

Subroutine Shot

C
C

```

DIMENSION TR(550),A(550)
DIMENSION REF(550),BKG(550),CHAN(550)
CHARACTER*2 Num,Prefix, BkgPrefix,Abs
Integer RefTrack,Start,End,Count,Hi,Lo
CHARACTER*4 Nfile, Outfile, Bkgfile
CHARACTER*1 LoC,HiC

```

```

C
C
50    Call Clear
      Type 100
100   FORMAT(///,20X,'SHOT DATA ANALYSIS',/)
C
190   TYPE 200
200   FORMAT($,///,'Input name of SHOT data file or mask: '
      *' (? for Directory, Q to Quit): ')
      READ(5,1200,ERR=50)Prefix
      IF (Prefix.EQ.'?') THEN
          Call Directory
      ELSEIF (Prefix.EQ.'Q') THEN
          Return
      ENDIF
      IF (Prefix.EQ.'?') GOTO 50

C
C
390   TYPE 400
400   FORMAT($,///,'Input name of BKG data file or mask: '
      *' (? for Directory, Q to Quit): ')
      READ(5,1200,ERR=390)BkgPrefix
      IF (BkgPrefix.EQ.'?') THEN
          Call Directory
      ELSEIF (BkgPrefix.EQ.'Q') THEN
          Return
      ENDIF
      IF (BkgPrefix.EQ.'?') GOTO 390

C
520   TYPE 530
530   FORMAT($,///,'Input name of ABS data storage file/mask: ')
      READ(5,1200,ERR=520)ABS

C
C
C
700   Type 700
      Format($,///,'Which track is to be used for the'
      *' reference data: ? ')
      Read(5,1300,Err=9999)RefTrack

C
D     PRINT*,'RefTrack = ',RefTrack

      Hi = int(RefTrack/10)
      Lo = RefTrack - Hi * 10
      HiC = CHAR(Hi+48)
      LoC = CHAR(Lo+48)

```

```

Num = HiC//LoC
Nfile = Prefix//Num
C
D PRINT*, 'Hi,Lo,Hic,Loc,Num,Nfile=',Hi,Lo,HiC,LoC,Num,Nfile
OPEN(UNIT=9,FILE=NFILE,STATUS='OLD',ERR=9999)
DO I=1,500
    READ(9,2000)CHAN(I),REF(I)
ENDDO
CLOSE(UNIT=9)

C
C
800 Type 810
810 Format($,/, 'Input start track: ')
    Read(5,1000,Err=800)Start
820 Type 830
830 Format($,/, 'Input end track: ')
    Read(5,1000,Err=820)End
    If(Start.GT.End) GOTO 800
    If (START.LT.RefTrack) GOTO 800
    If ((Start.LT.1).OR.(End.GT.50)) Goto 800

C
Do Count = Start,End
    Hi = int(Count/10)
    Lo = Count - Hi * 10
    Num = CHAR(Hi+48)//CHAR(Lo+48)
    Nfile = Prefix//Num
    Bkgfile = BkgPrefix//Num
    Outfile = Abs//Num
D PRINT*, 'Bkgfile, Outfile,Nfile=',Bkgfile,Outfile,Nfile
OPEN(UNIT=9,FILE=NFILE,STATUS='OLD')
OPEN(UNIT=8,File=Bkgfile,Status='OLD')
DO J=1,500
    READ(9,2000)CHAN(J),TR(J)
    READ(8,2000)CHAN(J),BKG(J)
ENDDO

C
CLOSE(UNIT=9)
CLOSE(UNIT=8)

C
C
C
C Calculate Absorbance
DO J=1,500
    T0 = REF(J) - BKG(J)
    IF (T0.LE.0.0) THEN
        T0 = 1.0
    ENDIF
    T = TR(J) - BKG(J)
    IF (T.LE.0.0) THEN
        T = 1.0
    ENDIF
    TT = T0/T
    A(J) = ALOG10(T0/T)
ENDDO

```

C

```
OPEN(UNIT=9,FILE=OUTFILE,STATUS='NEW',ERR=9999)
Write(6,910)Outfile
910  Format($,/,20x,'Creating Delta Absorbance file ',A4)
      DO I=1,500
          WRITE(9,2000)CHAN(I),A(I)
      ENDDO
      CLOSE(UNIT=9)
ENDDO
```

C

```
9998  GOTO 9997
1000  FORMAT(I3)
1300  Format(I2)
1200  FORMAT(A)
2000  FORMAT(1p,2e15.6)
9999  Print*,'Error has occurred...'
9997  Return
END
```

(18) *SimpleMath*: Performs simple mathematical manipulations (addition, subtraction, division, multiplication) on two data files.

Subroutine Simplemath(k)

```
real*4 f(4,512)
integer i,j,k,ntrack(2)
character*40 file(3)

1      Call Clear
      Type 5000
5000  Format(//,20X,'Mathematical Operations',//)
      do 8 i=1,2
2      write(6,1000)i
      read(5,1010,err=6)file(i)
      IF (file(I).EQ.'?') Then
          Call Directory
      ELSEIF (file(I).EQ.'Q') Then
          RETURN
      ENDIF
      IF (file(I).EQ.'?') THEN
          Goto 1
      Endif
      open(unit=i,file=file(i),err=7,status='old')
      goto 8

6      write(6,1020)
      goto 2

7      write(6,1030)
      goto 2
```

```

8      continue
      goto 10
10     do 20 i=1,2
        write(6,1035)i
        read(i,*,err=15,end=15) (f(4,j),f(i,j),j=1,512)
        goto 18
15     ntrack(i)=j-1
18     close(i)
20     continue
25     if (ntrack(1).eq.ntrack(2)) goto 28
        write(6,1040)
        goto 1
28     goto(100,110,120,130),k
100    do 105 j=1,ntrack(1)
        f(3,j)=f(1,j)+f(2,j)
105    continue
        goto 200
107    write(6,1100)
110    write(6,1060)
        read(5,1110,err=107)k
        i=abs(k-3)
        if ((k.eq.1).or.(k.eq.2)) goto 111
        goto 110
111    do 115 j=1,ntrack(1)
        f(3,j)=f(i,j)-f(k,j)
115    continue
        goto 200
120    do 125 j=1,ntrack(1)
        f(3,j)=f(1,j)*f(2,j)
125    continue
        goto 200
127    write(6,1100)
130    write(6,1061)
        read(5,1110,err=127)k
        i=abs(k-3)
        if ((k.eq.1).or.(k.eq.2)) goto 131
        goto 130
131    do 135 j=1,ntrack(1)
        f(3,j)=f(i,j)/f(k,j)
135    continue

```

```

        Goto 200
195    write(6,1020)
200    write(6,1070)
        read(5,1010,err=195)file(3)
        open(unit=3,status='new',err=210,file=file(3))
        goto 220
210    write(1030)
        goto 200
220    do 230 j=1,ntrack(1)
        write(3,1080)f(4,j),f(3,j)
230    continue
        close(3)

        Goto 9999
1000   format($,/'Enter name of input file number ',i1,'(? for'
        *' Directory, Q to Quit) : ')
1010   format(a40)
1020   format('/Invalid filename, please re-enter')
1030   format('/Unable to open file, please re-enter')
1035   format('/Loading file 'i1)
1040   format('/Files have unequal lengths, start over'//)
1060   format($,/'Enter file to be subtracted (1 or 2): ')
1061   format($,/'Enter file to be divisor (1 or 2): ')
1070   format($,/'Enter name of output file: ')
1080   format(1p,2e15.6)
1100   format('/Invalid response, please re-evaluate')
1110   format(i2)

9999   Return
        End

```

(19) *Smooth*: Smooths any data file.

```

        Subroutine Smooth

        real*4 f(4,512)
        integer i,j,k,l,m,ntrack(2)
        character*40 file(3)

150    Call Clear
        Type 5000
5000   Format(/,20X,'Smoothing Operation',//)
        write(6,1000),1
        read(5,1010,err=145)file(1)
        IF (file(1).EQ.'?') THEN

```

```

        Call Directory
ELSEIF (file(1).EQ.'Q') then
    Return
ENDIF
IF (file(1).EQ.'?') then
    GOTO 150
Endif
open(unit=1,status='old',err=160,file=file(1))
goto 170
160  write(6,1030)
     goto 150
170  write(6,1035)1
     read(1,*,err=175,end=175) (f(4,j),f(1,j),j=1,512)
     goto 180
175  ntrack(1)=j-1
180  close(1)
     goto 700
195  write(6,1020)
200  write(6,1070)
     read(5,1010,err=195)file(3)
     open(unit=3,status='new',err=210,file=file(3))
     goto 220
210  write(1030)
     goto 200
220  do 230 j=1,ntrack(1)
     write(3,1080)f(4,j),f(3,j)
230  continue
     close(3)

     Goto 9999
700  write(6,1160)
     read(5,*,err=695)k
     k=int(k/2)
     do 710 j=1,ntrack(1)
         f(3,j)=0
         if (j.le.k) then
             l=1
             m=2*k-j
             goto 703
         else
             l=j-k
         endif
     if (ntrack(1).le.(j+k)) then

```

```

        m=ntrack(1)
        l=2*j-ntrack(1)
        goto 703
    else
        m=j+k
    endif
703    do 705 i=1,m
        f(3,j)=f(3,j)+f(1,i)

705    continue
        f(3,j)=f(3,j)/(m-l+1)

710    continue
        goto 200

695    Write(6,1100)
145    Write(6,1020)

1000   format($,/'Enter name of input file number 'i1,' (? for
        *' Directory, Q to Quit) : ')
1010   format(a40)
1020   format('/Invalid filename, please re-enter')
1030   format('/Unable to open file, please re-enter')
1035   format('/Loading file 'i1)
1070   format($,/'Enter name of output file: ')
1080   format(1p,2e15.6)
1160   format($,/'Enter smooth window length in points: ')
1100   Format('/','Invalid response, please re-evaluate')

9999   Return
        End

```

(20) *Stplrg*: Used by the linear regression routines.

```

        SUBROUTINE STPLRG(M,X,Y,N,C,IFLAG,A)
C
C   THIS SUBROUTINE WILL FOR A POLYNOMIAL OF DEGREE N
C   TO M OBSERVATIONS, BY THE METHOD OF LEAST SQUARES.
C
        DIMENSION X(1), Y(1), A(1), C(1)
        N1 = N + 1
        N2 = N1 + 1
        NNPN = N*N1
        N1PN1 = N1 + N1
        N1N1 = NNPN + N1
C   CLEAR A ARRAY
        NA = (N + 1) * (N + 2)
        DO 5 IA=1,NA
            5   A(IA) = 0.0
C   COMPUTE SUMS OF POWERS
        DO 10 K = 1,M
            XPWR = X(K)

```

```

      I1J = N1 + 1
      DO 20 J = 2,N1
      JNP2 = N1N1 + J
      A(I1J) = A(I1J) + XPWR
      A(JNP2) = A(JNP2) + XPWR*Y(K)
      XPWR = XPWR*X(K)
20    I1J = I1J + N1
      DO 50 I = 2,N1
      IN1 = NNPN + I
      A(IN1) = A(IN1) + XPWR
50    XPWR = XPWR*X(K)
10    A(N1N1 + 1) = A(N1N1 + 1) + Y(K)
C    FILL IN REST OF MATRIX A
      A(1) = M
      DO 60 I = 2,N1
      IJ = I
      DO 60 J = 1,N
      IM1JP1 = IJ + N
      A(IJ) = A(IM1JP1)
60    IJ = IJ + N1
C    SOLVE EQUATIONS FOR COEFFICIENTS
      NC = -N2
      CALL MATINV(IFLAG,IFLAG,N1,NC,A,N1,I,XPWR)
      K = N1N1 + N1
      DO 70 I = 1,N1
      C(I) = A(K)
70    K = K - 1
      RETURN
      END
      SUBROUTINE MATINV(ISOL,IDSOL,NR,NC,A,MRA,KWA,DET)
C
C    THIS SUBROUTINE FINDS THE INVERSE AND/OR SOLVES
C    SIMULTANEOUS EQUATIONS, OR NEITHER, AND
C    CALCULATES A DETERMINANT OF A REAL MATRIX.
C
      DIMENSION A(1), KWA(1)
      IR = NR
      ISOL = 1
      IDSOL = 1
      IF(NR.LE.0) GO TO 330
      IF((IR-MRA).GT.0) GO TO 330
      IC = IABS(NC)
      IF ((IC - IR).LT.0) IC = IR
      IBMP = 1
      JBMP = MRA
      KBMP = JBMP + IBMP
      NES = IR*JBMP
      NET = IC*JBMP
      IF(NC) 10,330,20
10    MDIV = JBMP + 1
      IRIC = IR - IC
      GO TO 30
20    MDIV = 1
30    MAD = MDIV

```

```

MSER = 1
KSER = IR
MZ = 1
DET = 1.0
40  PIV = 0.
    I = MSER
50  IF (( I - KSER).GT.0) GO TO 70
    IF((ABS(A(I))-PIV).LE.0.) GO TO 60
    PIV = ABS(A(I))
    IP = I
60  I = I + IBMP
    GO TO 50
70  IF(PIV.EQ.0.) GO TO 340
    IF(NC.LT.0) GO TO 80
    I = IP-((IP - 1)/JBMP)*JBMP
    J = MSER - ((MSER - 1)/JBMP)*JBMP
    JJ = MSER/KBMP + 1
    II = JJ + (IP -MSER)
    KWA(JJ) = II
    GO TO 90
80  I = IP
    J = MSER
90  IF (IP - MSER) 330,120,100
100 IF ((J - NET).GT.0) GO TO 110
    PSTO = A(I)
    A(I) = A(J)
    A(J) = PSTO
    I = I + JBMP
    J = J + JBMP
    GO TO 100
110 DET = - DET
120 PSTO = A(MSER)
    DET = DET*PSTO
130 IF (DET.eq.0.) GOTO 150
140 PSTO = 1./PSTO
    GO TO 160
150 IDSOL = 1
    ISOL = 2
    RETURN
160 CONTINUE
    A(MSER) = 1.0
    I = MDIV
170 IF((I - NET).GT.0) GO TO 180
    A(I) = A(I)*PSTO
    I = I + JBMP
    GO TO 170
180 IF((MZ - KSER).GT.0) GO TO 210
    IF((MZ-MSER).EQ.0) GO TO 200
    I = MAD
    J = MDIV
    PSTO = A(MZ)
    IF(PSTO.EQ.0.) GO TO 200
    A(MZ) = 0.
190 IF((J-NET).GT.0) GO TO 200

```

```

        A(I) = A(I) - A(J)*PSTO
        J = J + JBMP
        I = I + JBMP
        GO TO 190
200    MAD = MAD + IBMP
        MZ = MZ + IBMP
        GO TO 180
        210 continue
C 210 NEED A TEST HERE.....CALL OVERFL(IVF)
C    GO TO (350,220),IVF
CCCCCCC NEED AT TEST HERE, ANYHOW
220    KSER = KSER + JBMP
        IF ((KSER-NES).GT.0) GO TO 260
        MSER = MSER + KBMP
        IF(NC.LT.0) GO TO 230
        MDIV = MDIV + IBMP
        MZ = ((MSER - 1)/JBMP)*JBMP + 1
        MAD = 1
        GO TO 40
230    MDIV = MDIV + KBMP
        IF(IRIC.NE.0) GO TO 240
        MZ = MSER + IBMP
        GO TO 250
240    MZ = ((MSER - 1)/JBMP)*JBMP + 1
250    MAD = MZ + JBMP
        GO TO 40
260    IF(NC.LT.0) RETURN
        JR = IR
270    IF(JR) 330,360,280
280    IF(KWA(JR) - JR) 330,320,290
290    K = (JR - 1)*JBMP
        J = K + IR
        L = (KWA(JR) - 1)*JBMP + IR
300    IF(J - K) 330,320,310
310    PSTO = A(L)
        A(L) = A(J)
        A(J) = PSTO
        J = J - IBMP
        L = L - IBMP
        GO TO 300
320    JR = JR - 1
        GO TO 270
330    ISOL = 3
        RETURN
340    DET = 0.
        ISOL = 2
        IDSOL = 1
        RETURN
350    ISOL = 2
        IDSOL = 2
360    RETURN
        END

```

(21) *Track*: Converts the ASCII data into intensity versus wavelength (or channels), while the track (or time) is held constant.

Subroutine Track

```
*          VARIABLE DEFINITIONS          *
integer ntrack,nchan,vtrack,i,j,decide,mcal,bcal
integer start, tend, count, bit(2)
character*20 filein, fileout
character*2 prefix
real*4 fdat(501,501),chan

10      data ntrack,nchan,mcal,bcal /500,500,1,0/

*          GET INPUT DATA          *

20      Call Clear
        Type 6000
6000    Format(//,20X,'Put data into Tracks')
        write(6,1070)
4999    Type 5000
5000    Format($,//,'Enter name of input file (? for'
*' Directory, Q to Quit): ')
        read(5,1000,err=21,end=2000)filein
        If (filein.eq.'?') then
            Call Directory
        ELSEIF (filein.EQ.'Q') Then
            Return
        ENDIF
        If (filein.eq.'?') then
            goto 4999
        endif
        open(unit=10,file=filein,status='old',err=22)
        goto 30

21      write(6,1010)
        goto 20

22      write(6,1020)
        goto 20

30      write(6,1090)filein
        do 40 i=1,501
            read(10,*,err=45,end=45) (fdat(i,j),j=1,500)

40      continue
        write(6,1070)
        Type 5100
5100    Format($,//'File does not comply with format. Please'
*' re-enter')
```

```

close(10)
goto 20

45  ntrack=i-1

50  write(6,1070)
    Type 5200
5200 Format(//Enter the calibration slope and offset for the',
    * 'translation: ',/, 'Wavelength = a * chan + b')
    Type 5300
5300 Format($,/, 'Slope = ')
    read(5,9999,err=51,end=2000)mcal
    Type 5400
5400 Format($,/, 'Intercept = ')
    read(5,9999,err=51,end=2000)bcac
9999 format(f20.6)
    goto 60

51  write(6,1010)
    goto 50

*           OPTIONS MENU           *

60  write(6,1070)
    Type 5500
5500 Format($,/'Do you wish to:',5X,'1) Copy a single track.',
    */,20X,'2) Copy a multiple tracks.')

    Type 5600
5600 Format($,/'Please enter 1 or 2: ')

70  read(5,*,err=71,end=2000)decide
    Goto 90

71  write(6,1010)
    goto70

90  Goto (100,3000),decide

100 write(6,1070)
    WRITE(6,5699)FILEIN
5699 FORMAT($,/'Your current data file is: ',A20,/)
    Type 5700
5700 Format($,/'Enter name of output file: ')
    read(5,1000,err=101,end=2000)fileout
    open(unit=2,file=fileout,status='new',err=102)
    goto 200

101 write(6,1010)
    goto 100

102 write(6,1020)
    goto 100

```

```

200  write(6,1040)ntrack
      read(5,*,end=2000,err=201)vtrack
      if ((vtrack .le. (ntrack)) .and. (vtrack .ge. 1))
      *then
        goto 210
      endif

201  write(6,1010)
      goto 200

210  do 220 i=1,nchan
        chan = i*mcal + bcal
        write(2,1050) chan,fdat(vtrack,i)
220  continue
      close(2)
      write(6,1080)fileout
      goto 2000

3000 write(6,1070)
      WRITE(6,5699)FILEIN
      Type 5800
5800 Format($,/'Enter name of output file mask (2 letters): ')
      read(5,900,err=3100,end=2000)prefix
      goto 3500

3100 write(6,1010)
      goto 3000

3500 write(6,1041)
      read(5,*,end=2000)start
      Write(6,1042)
      Read(5,*,End=2000)tend

      If ((start.ge.tend).or.(start.lt.1).or.(tend.gt.50))then
        Write(6,1010)
        goto 3000
      Endif

      Do count = start,tend
      bit(1) = int(count/10)
      bit(2) = count-bit(1)*10
      Fileout = prefix//CHAR (bit(1)+48)//CHAR (bit(2)+48)
      Open (unit=2,file=fileout,status='new')
      vtrack = count
        do i=1,nchan
          chan = i*mcal + bcal
          write(2,1050) chan,fdat(vtrack,i)
        enddo
      close(2)
      write(6,1080)fileout
      Enddo

      Goto 2000

```

```

900    format(a2)
1000   format(a20)
1010   format('Invalid entry, please re-enter')
1020   format('Unable to open named file, please re-enter')
1040   format('$,//Enter track number (1 - ',i3,'). ')
1041   format('$,//Enter starting track number: ')
1042   format('$,//Enter ending track number: ')
1050   format(1p,2e15.6)
1070   format(//)
1080   format(//,a40,'created')
1090   format(//LOADING ',a20)

2000   close(10)
       close(2)
       Return
       End

```

C.3.2 MAK-ABS-DATA Programs

The following routines are used in the program *MAK-ABS-DATA* program. This program calls the C routine *PLOT1* (also used in *PREFIT* and *ABSORPTION*). This program is for use on a CIT-414A graphics terminal.

```

C      Program MAKEDATA
C
C
C      To compile and link, the following command is used:
C      fc mak-abs-data.f graph.c -o mak-abs-data
C
C      Written by Kelly G. Casey, Ph.D
C      Shock Dynamics Laboratory
C      Washington State University
C
C      Jan. 1992
C
C      DIMENSION XX(550),X(550),Y(550),A(10)
C      DIMENSION REF(550),BKG(550),CHAN(550)
C      DIMENSION C1(4),C2(4),C3(4),WIDTH(4)
C      CHARACTER ANSWER*3, NFILE*64
C
C
C      PARAMETER (PI=3.1415926)
C
C      ***** INPUT PARAMETERS FOR GENERATING DATA *****
C
80     TYPE 81
81     FORMAT('      MAK - ABS - DATA Program',//,
*'This programs creates absorption data.'//)
90     TYPE 91
91     FORMAT('NOTE:Except for the NUMBER OF PEAKS, ALL',/,
*'numerical data input must have a decimal point.'//,

```

```

*To break out of the program at any time, hit DELETE.',
*///)
100 TYPE 110
110 FORMAT(,'Input # of absorption peaks (<4): ')
    READ(5,1000,ERR=100)NP
    IF ((NP.GT.4).OR.(NP.LT.1)) GOTO 100
120 TYPE 130
130 FORMAT(/,'Input the PEAK CHN # for each peak: ')
    DO I=1,NP
        WRITE(6,133)I
133     FORMAT(,'Input peak channel for peak #:',I2,' ')
        READ(5,1100,ERR=120)C2(I)
    ENDDO
C
140 TYPE 150
150 FORMAT(/,'Input the WIDTH (in chn #) of each peak: ')
    DO I=1,NP
        WRITE(6,155)I
155     FORMAT(,'Input width for peak #:',I2,' ')
        READ(5,1100,ERR=140)WIDTH(I)
    ENDDO
C
    DO I=1,NP
        C1(I) = C2(I) - 0.5*WIDTH(I)
        C3(I) = C2(I) + 0.5*WIDTH(I)
D    PRINT*,'I,C1,C2,C3 = ',I,C1(I),C2(I),C3(I)
    ENDDO
        F1 = PI / (2*(C2(1)-C1(1)))
        F2 = PI / (2*(C2(2)-C1(2)))
        F3 = PI / (2*(C2(3)-C1(3)))
        F4 = PI / (2*(C2(4)-C1(4)))
D    PRINT*,'F1,F2,F3,F4 = ',F1,F2,F3,F4
C
C    ***** Make Data *****
C
300 TYPE 350
350 FORMAT(/,'Please input AMPLITUDE values for each peak: ')
    DO I=1,NP
        WRITE(6,401)I
401     FORMAT(,'Input amplitude for peak #',I2,' ')
        READ(5,1400)A(I)
    ENDDO
    DO I=1,500
        IF ((I.GE.C1(1)).AND.(I.LE.C3(1))) THEN
            X1=A(1) * SIN(F1*(I-C1(1)))
        ELSE
            X1 = 0.0
        ENDIF
        IF ((I.GE.C1(2)).AND.(I.LE.C3(2))) THEN
            X2=A(2) * SIN(F2*(I-C1(2)))
        ELSE
            X2 = 0.0
        ENDIF
        IF ((I.GE.C1(3)).AND.(I.LE.C3(3))) THEN

```

```

        X3=A(3) * SIN(F3*(I-C1(3)))
    ELSE
        X3 = 0.0
    ENDIF
    IF ((I.GE.C1(4)).AND.(I.LE.C3(4))) THEN
        X4=A(4) * SIN(F4*(I-C1(4)))
    ELSE
        X4 = 0.0
    ENDIF
    X(I) = X1 + X2 + X3 + X4
D    PRINT*,I,X1,X2,X3,X4,X(I) = ',I,X1,X2,X3,X4,X(I)
ENDDO

C
210  TYPE 220
220  FORMAT($, 'Add a baseline ? (DC-Offset): Y or N: ')
    READ(5,1200,ERR=210)ANSWER
    IF (ANSWER.EQ.'N') GOTO 250
230  TYPE 240
240  FORMAT ($, 'Input baseline value: ')
    READ(5,1400)BASELINE
    DO I=1,500
        X(I) = X(I) + BASELINE
    ENDDO

C
250  TYPE 260
260  FORMAT($, 'Add NOISE to the data: Y or N: ')
    READ(5,1200,ERR=210)ANSWER
    IFLAG = 0
    IF (ANSWER.EQ.'N') GOTO 169
261  TYPE 262
262  FORMAT($, 'Please input % of noise to add: (10%, 5%, etc) ')
    READ(5,1100,ERR=9999)PERCENT
    IFLAG = 1
    DO I=1,500
        RNDD = ((-1)**INT(10*RAN1(I)))
        RND = RNDD * (RAN1(I*I)/(100./PERCENT))
        X(I) = X(I) + (RND*X(I))
D    PRINT*,I,RND,X = ',I,RND,X(I)
    ENDDO

C
C    Create REF data & BKG data
169  IF (IFLAG.EQ.1) THEN
        DO I=1,500
            RNDD = ((-1)**INT(10*RAN1(I)))
            RND = RNDD * (RAN1(I*I)/(100./PERCENT))
            REF(I)= 2000. + (RND*2000.)
            BKG(I) = 200. + (RND*200.)
        ENDDO
    ELSE
        DO I=1,500
            REF(I) = 2000.
            BKG(I) = 200.
        ENDDO
    ENDIF

```

```

C
C      Must add BKG to data & REF - needed especially for scaling
      DO I=1,500
          X(I) = X(I) + BKG(I)
          REF(I) = REF(I) + BKG(I)
      ENDDO

C
C      Calculate TRANSMISSION spectra now
C      MUST scale ABS spectra to be between 0 & 1
C
      XMAX = -10000.
      XMIN = 10000.
      DO I=1,500
          IF (X(I).LT.XMIN) XMIN = X(I)
          IF (X(I).GT.XMAX) XMAX = X(I)
      ENDDO
      FACTOR = 0.9999/XMAX
      DO I=1,500
          XX(I) = X(I)
          X(I) = X(I) * FACTOR
      ENDDO
      DO I=1,500
          Y(I) = REF(I) * 10**(-X(I))
      ENDDO

C
C
C      Must scale TRAN data so Tran-min = Ref
C
      YMAX = -10000.
      YMIN = 10000.
      DO i=1,500
          IF (Y(I).LT.YMIN) YMIN = Y(I)
          IF (Y(I).GT.YMAX) YMAX = Y(I)
      ENDDO
      FACTOR1 = REF(250)/YMAX
      DO I=1,500
          Y(I) = Y(I) * FACTOR1
      ENDDO

C
C      Now, input wavelength calibration slope & intercept
C      For time being, will just input values for
C      shot #91-053
C      Slope = 5.856855 & Intercept = 2652.516 angstroms
C
      DO I=1,500
          CHAN(I) = I*5.856855 + 2652.516
      ENDDO

C
C      GRAPHICS
C
600     TYPE 610
610     FORMAT(//,'You are about to see a plot of your '
*        ' REF, BKG, & Transmission data.')
        TYPE 620

```

```

620  FORMAT(/, 'Hit RETURN when finished viewing ...')
      DO I=1,550000
          ZZZ = I
      ENDDO
625  CALL PLOT1(CHAN,REF,500,'Wavelength (angstroms)',
      *'Intensity',TR, REF, & BKG Data',2000.,6000.,
      * 250.,500.,0.,3000.,250.,500.,0,0,0)
      CALL PLOT1(CHAN,BKG,500,'Wavelength (angstroms)',
      * 'Intensity',TR, REF, & BKG Data',2000.,6000.,
      * 250.,500.,0.,3000.,250.,500.,0,0,0)
      CALL PLOT1(CHAN,Y,500,'Wavelength (angstroms)',
      * 'Intensity',TR, REF, & BKG Data',2000.,6000.,
      * 250.,500.,0.,3000.,250.,500.,2,0,1)
630  TYPE 640
640  FORMAT(/, 'Is this data OK? ',/, 'Type 1 if OK' /,
      *'Type 2 if not OK' /, 'Type 3 to see again.' /)
      TYPE 641
641  FORMAT($,/, 'Please input 1,2, or 3: ')
      READ(5,1200,ERR=630)ANSWER
      IF (ANSWER.EQ.'1') THEN
          GOTO 650
      ELSEIF (ANSWER.EQ.'2') THEN
          GOTO 100
      ELSEIF (ANSWER.EQ.'3') THEN
          GOTO 625
      ENDIF

C
C
650  TYPE 660
660  FORMAT(/, 'You are about to see a plot of your '
      * 'Absorption data.')
      TYPE 670
670  FORMAT(/, 'Hit RETURN when finished viewing ...')
      DO I=1,550000
          ZZZ = I
      ENDDO
680  CALL PLOT1(CHAN,XX,500,'Wavelength (angstroms)',
      *'Absorbance', 'Absorption Data',2000.,6000.,
      * 250.,500.,0.,3000.,250.,500.,2,0,1)
690  TYPE 700
700  FORMAT(/, 'Is this data OK? ',/, 'Type 1 if OK' /,
      *'Type 2 if not OK' /, 'Type 3 to see again.' /)
      TYPE 701
701  FORMAT($,/, 'Please input 1,2, or 3: ')
      READ(5,1200,ERR=630)ANSWER
      IF (ANSWER.EQ.'1') THEN
          GOTO 170
      ELSEIF (ANSWER.EQ.'2') THEN
          GOTO 100
      ELSEIF (ANSWER.EQ.'3') THEN
          GOTO 625
      ENDIF

C

```

```

170 TYPE 180
180 FORMAT($,/, 'Write ABS Data to a file for IGRAPH ? Y or N ')
    READ(5,1200,ERR=170)ANSWER
    IF (ANSWER.EQ.'N') GOTO 9999
190 TYPE 200
200 FORMAT($, 'Input name of file: ')
    READ(5,1200,ERR=190)NFILE
    OPEN(UNIT=9,FILE=NFILE,STATUS='NEW',ERR=9999)
    DO I=1,500
        WRITE(9,2000)CHAN(I),XX(I)
    ENDDO
    CLOSE(UNIT=9)

C
C
370 TYPE 380
380 FORMAT($,/, 'Write REF data to a file for IGRAPH ? Y or N ')
    READ(5,1200,ERR=370)ANSWER
    IF (ANSWER.EQ.'N') GOTO 400
390 TYPE 391
391 FORMAT($, 'Input name of file: ')
    READ(5,1200,ERR=370)NFILE
    OPEN(UNIT=9,FILE=NFILE,STATUS='NEW',ERR=9999)
    DO I=1,500
        WRITE(9,2000)CHAN(I),REF(I)
    ENDDO
    CLOSE(UNIT=9)

C
C
400 TYPE 410
410 FORMAT($,/, 'Write BKG data to a file for IGRAPH ? Y or N ')
    READ(5,1200,ERR=400)ANSWER
    IF (ANSWER.EQ.'N') GOTO 500
420 TYPE 430
430 FORMAT($, 'Input name of file: ')
    READ(5,1200,ERR=420)NFILE
    OPEN(UNIT=9,FILE=NFILE,STATUS='NEW',ERR=9999)
    DO I=1,500
        WRITE(9,2000)CHAN(I),BKG(I)
    ENDDO
    CLOSE(UNIT=9)

C
C
500 TYPE 510
510 FORMAT($,/, 'Write TRANSMISSION data to a file for '
    *' IGRAPH? Y or N ')
    READ(5,1200,ERR=500)ANSWER
    IF (ANSWER.EQ.'N') GOTO 9999
520 TYPE 530
530 FORMAT($, 'Input name of file: ')
    READ(5,1200,ERR=520)NFILE
    OPEN(UNIT=9,FILE=NFILE,STATUS='NEW',ERR=9999)
    DO I=1,500
        WRITE(9,2000)CHAN(I),Y(I)
    ENDDO

```

```

CLOSE(UNIT=9)
C
1000 FORMAT(I2)
1100 FORMAT(F5.2)
1200 FORMAT(A)
2000 FORMAT(2e15.6)
1310 FORMAT(F10.1,1X,F20.10)
1400 FORMAT(F10.3)
9999 CALL EXIT
      END

      FUNCTION RAN1(IDUM)
      Random number generator
C
C
C      From Numerical Recipes, pg 197
C
C      Return number between 0.0 & 1.0
C
C
      DIMENSION R(97)
      PARAMETER (M1=259200, IA1=7141,IC1=54773,RM1=1./M1)
      PARAMETER(M2=134456,IA2=8121,IC2=28411,RM2=1./M2)
      PARAMETER(M3=243000,IA3=4561,IC3=51349)
      DATA IFF /0/
      IF (IDUM.LT.0.OR.IFF.EQ.0) THEN
          IFF = 1
          IX1 = MOD(IC1-IDUM,M1)
          IX1=MOD(IA1*IX1+IC1,M1)
          IX2=MOD(IX1,M2)
          IX1=MOD(IA1+IX1+IC1,M1)
          IX3=MOD(IX1,M3)
          DO J=1,97
              IX1=MOD(IA1*IX1+IC1,M1)
              IX2=MOD(IA2*IX2+IC2,M2)
              R(J)=(FLOAT(IX1)+FLOAT(IX2)*RM2)*RM1
          ENDDO
          IDUM = 1
      ENDIF
      IX1=MOD(IA1*IX1+IC1,M1)
      IX2=MOD(IA2*IX2+IC2,M2)
      IX3=MOD(IA3*IX3+IC3,M3)
      J = 1 + (97*IX3)/M3
      IF (J.GT.97.OR.J.LT.1) PAUSE
      RAN1 = R(J)
      R(J) = (FLOAT(IX1)+FLOAT(IX2)*RM2)*RM1
      RETURN
      END

```

Table C.1: *ABSFIT* Routines

NAME	Language	Called By:	Purpose
Ambient	Fortran	Menu	Calculates the absorbance of a single data file referenced to another file.
Channel	Fortran	Menu	Convert the ASCII data into intensity versus track data. Wavelength or channel # is kept constant.
Clear	C	All other programs.	Clears the terminal screen
Correction	Fortran	Menu	Reflection correction routine for individual files.
Directory	C	All other programs.	Lists the files in the current directory.
Graph	C	Menu	A system call to the program IGRAPH.
Help	Fortran	Menu	A help screen telling the user about the program.
Linfit	Fortran	Menu	Routine to calculate wavelength shifts in absorption spectra.
Menu	Fortran	----	The main routine. All other routines are called by Menu.
Multiambient	Fortran	Menu	Calculates the absorbance of many data files referenced to other files.

Table C.1: *ABSFIT* Routines - Continued

NAME	Language	Called By:	Purpose
Multicorrect	Fortran	Correction	Reflection correction routine for many files.
Multismooth	Fortran	Menu	Smooths multiple data file.
OmaConvert	C	Menu	Converts the OMA binary data into ASCII data.
Plot1	C	Shifted	Graphics routine for the CIT-414A terminal.
Regress	Fortran	Linfit	Linear regression routine.
Shell	C	Menu	A system call, allowing the user to do any system command (i.e., change directories, make directories, etc.).
Shot	Fortran	Menu	Calculates the change in absorbance. Later time data files are referenced to earlier time files.
SimpleMath	Fortran	Menu	Does simple file manipulations. Allows the user to add, subtract, multiply, or divide any two data files.
Smooth	Fortran	Menu	Smooths any data file.
Stplrg	Fortran	Regress	Part of the linear regression routine.
Track	Fortran	Menu	Converts the ASCII data into intensity versus wavelength data. (Time or track number is kept constant).

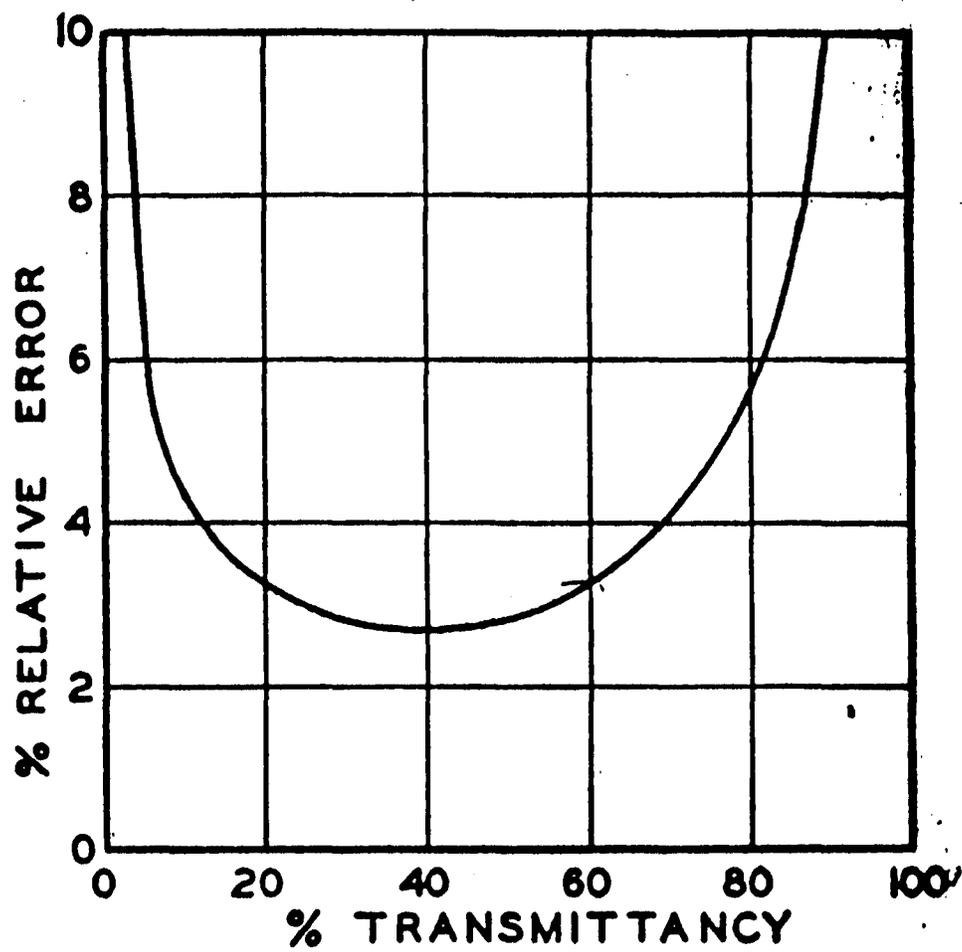


Figure C.1: Relative analysis error as a function of transmittancy.

Taken from reference #26.

Appendix D: Error Propagation

In general, if $z = z(x,y)$, then the propagation of error in z is^{19,30}:

$$\sigma_z = \sqrt{\left(\frac{\partial z}{\partial x}\right)^2 \sigma_x^2 + \left(\frac{\partial z}{\partial y}\right)^2 \sigma_y^2} \quad \text{D.1}$$

where σ_z is the standard error in z , σ_x is the standard error in x , σ_y is the standard error in y . If z is the absorbance ($A = \log(I_0/I)$), x is I_0 , and y is I , then the error in the absorbance is:

$$\sigma_A = \sqrt{\left(\frac{\partial A}{\partial I_0}\right)^2 \sigma_{I_0}^2 + \left(\frac{\partial A}{\partial I}\right)^2 \sigma_I^2} \quad \text{D.2}$$

The partial derivatives in D.2 are evaluated as:

$$\frac{\partial z}{\partial I_0} = \frac{\partial}{\partial I_0} \log\left(\frac{I_0}{I}\right) = \log(e) \frac{I}{I_0} \left(\frac{1}{I}\right) = \log(e) \frac{1}{I_0} \quad \text{D.3}$$

and as

$$\frac{\partial z}{\partial I} = \frac{\partial}{\partial I} \log\left(\frac{I_0}{I}\right) = \log(e) \frac{I}{I_0} (I_0) \left(\frac{-1}{I^2}\right) = -\log(e) \frac{1}{I} \quad \text{D.4}$$

Placing D.3 and D.4 into D.2 gives:

$$\sigma_z = \sqrt{\left(\log(e) \frac{1}{I_0}\right)^2 (\sqrt{I_0})^2 + \left(-\log(e) \frac{1}{I}\right)^2 (\sqrt{I})^2} \quad \text{D.5}$$

Here, the uncertainty in I and I₀ is taken as the squareroot of the vidicon counts. After manipulation, equation D.5 is:

$$\sigma_z = \log(e) \sqrt{\frac{1}{I_0} + \frac{1}{I}} \quad \text{D.6}$$

This absorbance error is dependent upon the reference and sample counts and therefore constantly changing. Earlier in this report (section 4.0) it was stated that the maximum error is approximately 10%. Using this as the counting uncertainty in equation D.5 yields the maximum absorbance error:

$$\sigma_z = \sqrt{\left(\log(e) \frac{1}{I_0}\right)^2 (0.1I_0)^2 + \left(-\log(e) \frac{1}{I}\right)^2 (0.1I)^2} \quad \text{D.7}$$

$$\sigma_z = \log(e) \sqrt{(0.1)^2 + (0.1)^2} \quad \text{D.8}$$

or

$$\sigma_z = 0.061 \approx 6\% . \quad \text{D.9}$$

The maximum error in any absorbance calculation is 6%.