

AD-A279 099



ESL-TR-90-47

2

SPECTROSCOPIC STUDIES OF HAZARDOUS FUEL INTER-ACTIONS WITH SOILS

T.L. TIPTON, DR D.E. STONE

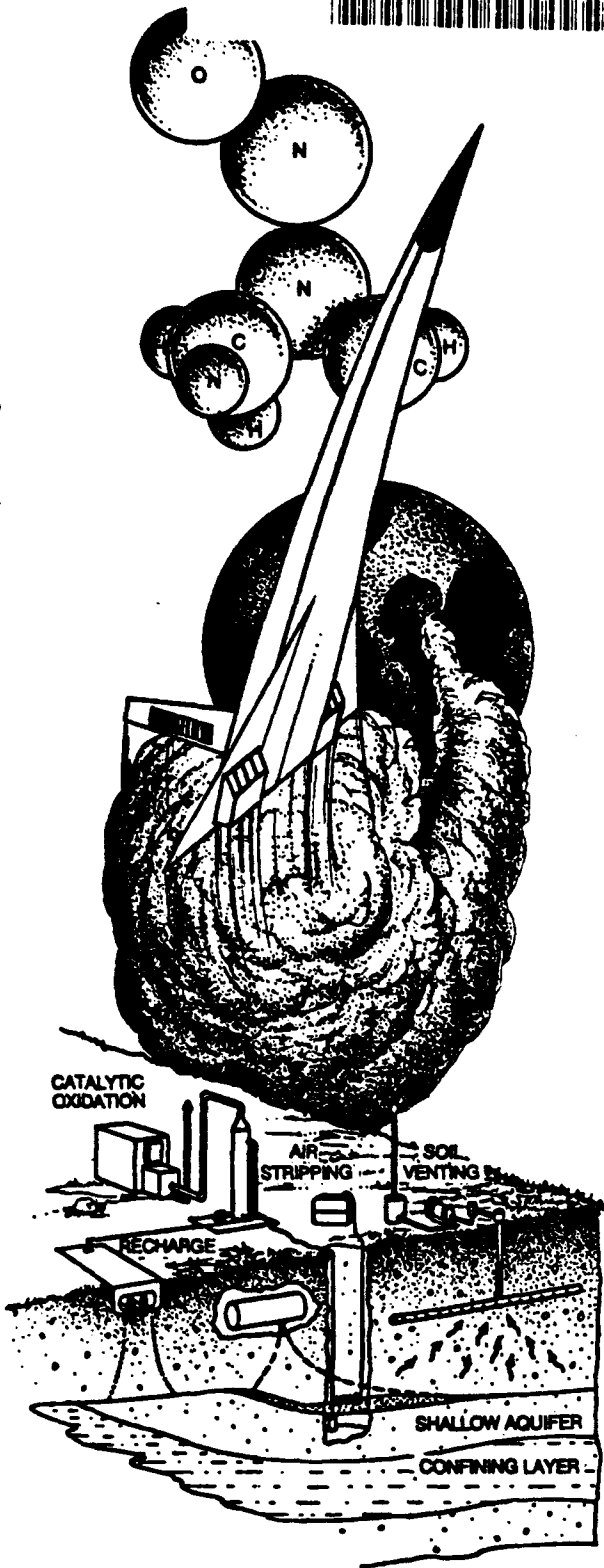
AIR FORCE ENGINEERING AND SERVICES CENTER
HQ AFESC/RDVC
TYNDALL AFB FL 32403-6001

DECEMBER 1993

FINAL REPORT

OCTOBER 1989 - OCTOBER 1990

DTIC
SELECTE
MAY 1 1994
S B D



APPROVED FOR PUBLIC RELEASE:
DISTRIBUTION UNLIMITED

94-13968



ENVIRONICS DIVISION
Air Force Engineering & Services Center
ENGINEERING & SERVICES LABORATORY
Tyndall Air Force Base, Florida 32403



94 5 00 059

DTIC QUALITY ASSURED 1

NOTICE

PLEASE DO NOT REQUEST COPIES OF THIS REPORT FROM
HQ AFESC/RD (ENGINEERING AND SERVICES LABORATORY).

ADDITIONAL COPIES MAY BE PURCHASED FROM:

NATIONAL TECHNICAL INFORMATION SERVICE
5285 PORT ROYAL ROAD
SPRINGFIELD, VIRGINIA 22161

FEDERAL GOVERNMENT AGENCIES AND THEIR CONTRACTORS
REGISTERED WITH DEFENSE TECHNICAL INFORMATION CENTER
SHOULD DIRECT REQUESTS FOR COPIES OF THIS REPORT TO:

DEFENSE TECHNICAL INFORMATION CENTER
CAMERON STATION
ALEXANDRIA, VIRGINIA 22314

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED		1b RESTRICTIVE MARKINGS	
2a. SECURITY CLASSIFICATION AUTHORITY		3 DISTRIBUTION / AVAILABILITY OF REPORT Approved for Public Release Distribution Unlimited	
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE			
4. PERFORMING ORGANIZATION REPORT NUMBER(S) ESL-TR-90- 47		5. MONITORING ORGANIZATION REPORT NUMBER(S)	
6a. NAME OF PERFORMING ORGANIZATION Air Force Engineering and Services Center	6b. OFFICE SYMBOL (If applicable) RDVC	7a. NAME OF MONITORING ORGANIZATION Air Force Engineering and Services Center	
6c. ADDRESS (City, State, and ZIP Code) HQ AFESC/RDVC Tyndall AFB FL 32403-6001		7b. ADDRESS (City, State, and ZIP Code) HQ AFESC/RDVC Tyndall AFB FL 32403-6001	
8a. NAME OF FUNDING / SPONSORING ORGANIZATION HQ AFESC	8b OFFICE SYMBOL (If applicable) RDV	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER In-House	
8c. ADDRESS (City, State, and ZIP Code) HQ AFESC/RDVC Tyndall AFB FL 32403-6001		10. SOURCE OF FUNDING NUMBERS	
		PROGRAM ELEMENT NO 6.1	PROJECT NO 0100
		TASK NO 8328	WORK UNIT ACCESSION NO. N/A
11. TITLE (Include Security Classification) Spectroscopic Studies of Hazardous Fuel Interactions with Soils			
12. PERSONAL AUTHOR(S) Terence L. Tipton and Daniel A. Stone			
13a. TYPE OF REPORT Interim	13b. TIME COVERED FROM 10/89 TO 10/90	14. DATE OF REPORT (Year, Month, Day) December 1993	15. PAGE COUNT
16. SUPPLEMENTARY NOTATION Availability of this report is specified on reverse of the front cover			
17. COSATI CODES		18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)	
FIELD	GROUP	Montmorillonite clay, infrared spectroscopy, ultraviolet-visible spectroscopy, fuels, soil chemistry, extraction.	
07	04		
17	05		
19. ABSTRACT (Continue on reverse if necessary and identify by block number) <p>This report describes the development and application of techniques for the spectroscopic study of hazardous fuel compounds in soils. The objective is to obtain improved models of bulk pollutant transport through soils. Two approaches have been used for this study: one consists of in-situ measurements of electronic and vibrational interactions via ultraviolet-visible spectroscopy and infrared spectroscopy, respectively; the other incorporates analytical separation techniques to extract contaminants from soils and identify individual reaction products. Initially, montmorillonite clay has been chosen as a prototype soil sample because the structure of this clay is well characterized and spectroscopic results can be readily interpreted. Further simplification has been obtained by exposing the clay samples to individual fuel components rather than mixtures. An interpretation of one clay/fuel system has been accomplished. The plan is to gradually increase the complexity of the samples until typical field conditions can be modeled.</p>			
20 DISTRIBUTION / AVAILABILITY OF ABSTRACT <input checked="" type="checkbox"/> UNCLASSIFIED/UNLIMITED <input type="checkbox"/> SAME AS RPT <input type="checkbox"/> DTIC USERS		21 ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED	
22a NAME OF RESPONSIBLE INDIVIDUAL Terence L. Tipton		22b TELEPHONE (Include Area Code) 904-283-6057	22c. OFFICE SYMBOL RDVC

EXECUTIVE SUMMARY

A. OBJECTIVE: Determine the fate of aromatic fuel compounds in soils via molecular-level measurements, and thereby obtain improved models of bulk pollutant transport through soils. These models would enhance the ability of the Air Force to contain and clean up fuel spills.

B. BACKGROUND: Current models of pollutant transport through soils are based entirely on bulk measurements (e.g. grams of chemical sorbed per gram of soil). These models effectively describe pollutant transport in some systems, but they cannot be reliably extrapolated to new systems unless they are founded on correct molecular-level assumptions.

C. SCOPE: A study was conducted on electronic and vibrational interactions of fuel compounds with montmorillonite clay. Montmorillonite was chosen as a prototype soil sample because the structure of this clay is well characterized and spectroscopic results can be readily interpreted. Further simplification was obtained by exposing clays to individual fuel components rather than mixtures. Both physical and chemical changes of montmorillonite were monitored as contamination proceeded in a controlled environment. Analytical separation techniques were used to extract contaminants from the montmorillonite samples and to identify individual reaction products.

D. METHODOLOGY: Ultraviolet-visible and infrared spectroscopy were used to obtain electronic and vibrational information, respectively. The former measures electronic properties of molecules, particularly the transformation of neutral organic molecules in soils to ions. The latter measures chemical changes via changes in molecular vibrations.

Extractions of clays were performed by repeatedly exposing the clays to common solvents such as acetone and chloroform. Separation of reaction products was effected by gas chromatography. Reaction products were then identified using either infrared spectroscopy or mass spectroscopy in conjunction with spectral library searches.

E. TEST DESCRIPTION: Some tests were conducted in a controlled-environment cell containing a self-supporting clay film (which was thin enough to transmit light), a vapor-phase aromatic contaminant, and a desiccant. This setup enabled clay/pollutant interactions to be studied as a function of relative humidity. The cell was transported back and forth between an ultraviolet-visible spectrometer and an infrared spectrometer to obtain tandem measurements of the clay at various stages of desiccation.

Other tests were conducted using a reflux apparatus containing about 1 gram of clay and about 70 mL of a pure liquid contaminant. The clay was refluxed overnight and then extracted in a commercial Soxhlet extraction apparatus. The extract was separated into its constituent compounds via a commercial gas chromatograph and analyzed with either a mass spectrometer or a Fourier transform infrared spectrometer.

F. RESULTS: A characterization of a simple system was accomplished: that of para-dimethoxybenzene on montmorillonite. This system was chosen as a starting point because the spectroscopic results could be readily interpreted and, therefore, the techniques for studying this system could be efficiently developed.

G. CONCLUSIONS: An initial step has been taken toward understanding the behavior of fuel contaminants in soils. Once some simple clay/pollutant systems have been interpreted, it should be possible to predict the behavior of similar systems. The

complexity of the systems will then be increased to a level representative of that existing at Air Force fuel facilities.

Parallel studies are underway at the University of Florida using a variety of techniques that cannot readily be implemented at Tyndall AFB. Complementary studies of bulk transport of pollutants through soils are in progress at Tyndall AFB. This multifaceted approach should improve the ability of the Air Force to minimize the environmental impact of fuel spills.

H. RECOMMENDATIONS: Future research should be directed towards increasing the complexity of the contaminated soil systems to a level comparable to field samples. This will likely require the implementation of the supercritical fluid extraction technique since conventional solvent extractions have been found to be of limited usefulness in removing pollutants from clays.

Studies on contaminated clays might lead to the development of an artificially modified clay that efficiently traps any aromatic molecules that sorb onto it. Such a clay could be used to contain a fuel spill within a localized area.

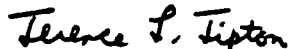
Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By _____	
Distribution/ _____	
Availability Codes	
Dist	Avail and/or Special
A-1	

PREFACE

This report was prepared by the Engineering and Services Laboratory of the Air Force Engineering and Services Center (AFESC), Tyndall Air Force Base FL 32403-6001. This work was sponsored by AFESC. Dr. Terence L. Tipton and Dr. Daniel A. Stone (AFESC/RDVC) were the government project officers. This report summarizes work accomplished between October 1989 and October 1990 under program element 61101F.

This report has been reviewed by the Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public including foreign nationals.

This technical report has been reviewed and is approved for publication.



TERENCE L. TIPTON, GS-12
Research Chemist



NEIL J. LAMB, Colonel, USAF, BSC
Chief, Environics Division



HOWARD R. MEYER Jr., Maj, USAF
Chief, Environmental Interactions
R&D Branch

TABLE OF CONTENTS

Section	Title	Page
I	INTRODUCTION.....	1
	A. OBJECTIVE.....	1
	B. BACKGROUND.....	1
	C. SCOPE.....	4
II	EXPERIMENTAL PROCEDURES.....	6
	A. CLAY FILM PREPARATION.....	6
	1. Starting Material.....	6
	2. Size Fractionation.....	6
	3. Ion Exchange.....	7
	4. Production of Clay Films.....	8
	B. COMMERCIAL INSTRUMENTATION.....	9
	C. CLAY-FILM CELL FOR TANDEM FT-IR AND UV-VISIBLE STUDIES.....	10
	D. CLAY EXTRACTION.....	12
	1. Sample Preparation.....	12
	2. Soxhlet Extraction.....	13
	3. Supercritical Fluid Extraction.....	16
	4. Sample Heater for Matrix Isolation...	18
	5. Apparatus for: Gas Chromatography/ Matrix-Isolation/FT-IR Spectroscopy..	21
III	RESULTS AND DISCUSSION.....	27
	A. TANDEM FT-IR AND UV-VISIBLE EXPERIMENTS ON CONTAMINATED CLAYS.....	27
	B. ANALYSIS OF CLAY EXTRACTS.....	34
	C. MATRIX-ISOLATION EXPERIMENTS WITH HEATED SAMPLES.....	34
IV	CONCLUSIONS.....	38
V	FUTURE PLANS.....	39
	REFERENCES.....	41

TABLE OF CONTENTS
(Concluded)

Section	Title	Page
APPENDIX.....		45
A.	COMPUTER PROGRAMS FOR THE NICOLET 660 WORK STATION.....	45
B.	COMPUTER PROGRAMS FOR THE PERKIN-ELMER 7500 WORK STATION.....	166

LIST OF FIGURES

Figure	Title	Page
1	Clay-Film Cell for Tandem FT-IR and UV-VIS Experiments.....	11
2	Apparatus for Refluxing Clay Under Nitrogen...	14
3	Soxhlet Extractor.....	15
4	Apparatus for Matrix Isolation of Supercritical Fluid Extracts.....	17
5	Sample Heater for Matrix-Isolation Experiments	19
6	GC/MI/FTIR Apparatus.....	23
7	FT-IR Spectrum of Cu-Exchanged Montmorillonite	28
8	Charge-Transfer Reaction for Benzene on Cu-Montmorillonite.....	30
9	FT-IR Spectra of Cu-Montmorillonite + DMOB + P ₂ O ₅ : A) at 0.0 hr with Only Clay Present, B) at 3.0 Hr, C) at 69 Hr, D) After Reintroduction of Ambient Air and Removal of DMOB.....	31
10	Growth of FT-IR and UV-VIS Bands of a DMOB/CUM Film as a Function of Elapsed Time of Dehydration: A) 0.1 Hr, B) 2.0 Hr, and C) 3.0 Hr.....	33
11	Sequential FT-IR Spectra of Toluene/CUM: A) CUM in Ambient Air, B) CUM After Refluxing in Toluene, and C) Acetone Extract of Toluene/CUM.....	35
12	GC/MS Chromatogram for Acetone Extract of Toluene/CUM.....	36

SECTION I
INTRODUCTION

A. OBJECTIVE

The objective of this project is to determine the mechanisms by which component compounds of Air Force fuels interact with soils, and thereby improve the predictive capabilities of macroscopic models of pollutant transport through soils. Current models of soil transport (e.g. see Reference 1) are derived from bulk measurements and lack a firm theoretical foundation based on molecular-level interactions.

B. BACKGROUND

This project is an in-house study in progress at Tyndall AFB under the category of Environmental Impact Assessment. Its purpose is to provide additional framework for concurrent in-house studies of bulk transport properties of aquifer pollutants. It is being supported by parallel research underway at the Soil Science Department of the University of Florida (UF) under a contract with the Air Force. The UF provides additional instrumentation and expertise necessitated by the complex nature of this project.

A number of different types of measurements are required to understand the nature of interactions of pollutants with soils. These can be divided into two categories: (1) those that are

performed directly on the soils; and (2) those that are performed on materials extracted from the soils. The former includes Fourier transform infrared (FT-IR) spectroscopy, ultraviolet-visible (UV-VIS) spectroscopy, and gravimetric measurements. The latter includes the combination of gas chromatography and mass spectrometry (GC/MS); and the combination of GC, matrix-isolation (MI), and FT-IR spectroscopy (GC/MI/FTIR).

Many of the above measurements can be made with equipment available at Tyndall AFB. Important supplemental information can be obtained from the wider range of facilities available at UF. This includes: (1) x-ray diffraction to monitor changes in interlayer spacing of soils that occurs on adsorption of fuel compounds; (2) MI spectroscopy of ions to probe the behavior of organic ions that form in soils; and (3) FT-Raman spectroscopy to selectively excite a specific molecular specie in a soil from among the many species that are likely to be present.

One approach that has been developed for studying polluted soil samples is to combine the techniques of UV-VIS and FT-IR spectroscopy. UV-VIS spectroscopy measures electronic properties of molecules, particularly the transformation of neutral organic molecules in soils to ions. FT-IR spectroscopy measures chemical changes via changes in molecular vibrations.

Spectroscopic absorption can be related to macroscopic properties by recording spectroscopic and gravimetric data simultaneously. A microbalance provides the sensitivity necessary

to measure sorption of pollutants onto the tiny quantities of sample that are used in spectroscopic measurements.

Initially, clays are being used as prototype soil samples because they: (1) have structures that are well characterized; (2) can be fashioned into self-supporting films that can be examined via conventional spectroscopic absorption measurements; (3) readily sorb atmospheric moisture, and are thus amenable to the study of reaction rate versus humidity; (4) have large interlayer surface areas that enable a relatively large quantity of contaminant sorption per gram of clay; and (5) are important components of natural soils.

In-situ measurements on clays have proven to be useful for the study of reactions on clay films, but these measurements are deficient in two areas. First, soils are opaque except when examined in very minute quantities. Consequently, the sensitivity of spectroscopic measurements is very limited. Second, reaction products are difficult to identify when several are simultaneously present in a soil sample. These deficiencies can be at least partially remedied by extracting soil samples with nonaromatic solvents such as chloroform and acetone. Reaction products can then be separated using standard analytical techniques, particularly GC. Identification of products can be achieved by introducing the effluent of a GC to either a mass spectrometer or an FT-IR spectrometer.

In addition to standard solvent extraction, supercritical fluid extraction (SFE) may be applicable to contaminated soils.

A supercritical fluid is a highly compressed gas whose temperature is too high to permit condensation to a liquid. Supercritical fluids are often highly effective solvents for extracting solutes from solid samples, since these fluids can easily penetrate microporous materials.

It is well known that aromatic molecules are susceptible to ionization when they are absorbed into the interlayers of clays (e.g., see Reference 2). Such ionization is induced by transition metal ions which occur naturally in the interlayers. Formation of aromatic ions is evidenced by color changes that are readily discernible to the naked eye. The aromatic ions often polymerize, particularly under dry conditions. An understanding of the reaction mechanisms has been achieved only for some very simple systems (e.g., see Reference 3). Accurate modeling of the behavior of pollutants in a natural soil environment will require much additional study.

C. SCOPE

Section II of this report begins with a procedure for preparing clay film samples for spectroscopic analyses. Next, it explains how contaminated clays are extracted for analyses of reaction products. Third, it describes the instrumentation used for examining clays and clay extracts.

Section III presents some of the results obtained by applying the experimental procedures of Section II. It includes a

description of tandem FT-IR/UV-VIS experiments on clays contaminated with components of Air Force fuels. It also includes results of extraction studies via GC/MS.

Section IV presents conclusions obtained from preliminary data, and Section V suggests how these results may be extended in the future.

The Appendix tabulates computer programs that were written to collect, organize, and plot data on the computer work stations for the FT-IR and UV-VIS spectrometers.

SECTION II
EXPERIMENTAL PROCEDURES

A. CLAY FILM PREPARATION

1. Starting Material

The starting material for all experiments was SAz-1 montmorillonite obtained from the Clay Minerals Society Source Clay Minerals Repository of the University of Missouri.

2. Size Fractionation

The starting material had to be size fractionated to obtain a pure clay. The procedure for this process is as follows:

a. Mix 30.0 grams of SAz-1 montmorillonite with 1.20 liters of 0.5 M NaCl to form a suspension. The NaCl facilitates the size-fractionation of the clay (described in Step h).

b. Distribute the suspension equally among six 250-mL centrifuge bottles.

c. Centrifuge the bottles at 3000 rpm for 10 minutes.

d. Decant and discard the supernatant from the bottles.

e. Add 200 mL of distilled deionized water to each bottle.

f. Shake the bottles in a commercial shaker at about 2000 vibrations per minute until uniform suspensions are obtained (typically 10 minutes).

g. Repeat the above washing procedure (Steps c-f) until the addition of silver nitrate solution to the wash gives no precipitate (typically, five washings are required). Increase the centrifuging times if any of the clay remains suspended after completion of Step c.

h. Remove the coarsest particles from the suspensions by centrifuging the bottles for a few minutes or by letting the suspensions sit undisturbed for approximately 30 minutes.

i. Decant the bottles and save only the decanted material.

j. Centrifuge the bottles at 3000 rpm for 10 minutes.

k. Decant and discard the supernatant from the bottles.

3. Ion Exchange

The clay samples were subjected to ion exchange to replace the ions which were naturally present in the clay with the ions which were targeted for study. The following procedure describes how Cu-exchanged montmorillonite is produced from the product of the size-fractionation procedure outlined above:

a. Add 200 mL of 0.05 M Cu(II)-chloride to each of the bottles.

b. Shake the bottles at 2000 vibrations per minute until uniform suspensions are obtained.

c. Centrifuge the bottles at 3000 rpm for 10 minutes.

d. Decant and discard the supernatants from the bottles.

e. Repeat Steps a-b.

f. Perform the washing procedure (Steps c-f of the size fractionation procedure) until the addition of silver nitrate solution to the wash gives no precipitate.

4. Production of Clay Films

a. Determine the approximate mg clay per mL of suspension by weighing a sample of the suspension before and after drying it. (Note: For the SAz-1 montmorillonite used in this report, yields of better than 50 percent were recorded for the size-fractionation step).

b. Dilute the clay suspension with distilled deionized water to obtain roughly 2 mg of clay per mL of suspension. If the concentration is too low, a self-supporting film cannot be produced. If the concentration is too high, the resulting film will absorb too much light to be useful for spectroscopic measurements.

c. Using a pipet, deposit several 1 mL samples of the final clay suspension onto a polyethylene sheet. Allow 18 hr for the samples to dry in the open air. If faster drying is needed, put the polyethylene sheet in a box equipped for a dry air flow.

d. Peel the dried clay samples from the polyethylene sheet by running the sheet over a knife edge. The dried clay films are pale green, roughly circular, and about 1.5 cm in diameter.

B. COMMERCIAL INSTRUMENTATION

The infrared measurements described in this report were recorded on a Nicolet Model 740 FT-IR spectrometer equipped with an HgCdTe-A detector (5000 to 600 cm^{-1}). The resolution was set at 0.5 cm^{-1} for MI experiments, and 2.0 cm^{-1} for all other experiments. The iris aperture for the source radiation was set at 2 mm. At least 100 averaged scans were recorded for each spectrum.

UV-VIS spectra were recorded on a Perkin-Elmer Model 3840 photodiode-array spectrophotometer at 1.5 nm resolution. Each spectrum consisted of 256 averaged scans covering the range from 190 nm to 900 nm.

GC/MS spectra were recorded on a Hewlett-Packard Model 5980A GC, coupled to a Model 5970 mass selective detector. A 1.0 microliter sample was injected via the splitless mode into a 30-meter long, 0.322 mm i.d. capillary coated with a 5 percent phenyl, 95 percent methyl, 0.25 micrometer polymethylsiloxane film (J&W Scientific). The oven was held at 40°C for 4 minutes, ramped to 250°C at 10°C minute^{-1} , and held at 250°C for 20 minutes. The carrier gas was helium, and the column head pressure was 15 psi above atmospheric pressure.

C. CLAY-FILM CELL FOR TANDEM FT-IR AND UV-VISIBLE STUDIES

A controlled-environment cell was constructed to permit analyses of clay films via tandem FT-IR and UV-VIS measurements. A 25 mm i.d. KF[®] tee was augmented with two additional arms to produce a cross having four horizontal arms and one vertical arm (Figure 1). This configuration enabled a pair of infrared transmission windows (KCl) and a pair of UV-VIS transmission windows (sapphire) to be mounted on the horizontal arms of the cell at the same time. The vertical arm was used to suspend a rotatable stainless steel sample holder in the path of either the FT-IR or UV-VIS beam. The extra horizontal arm was made shorter than the other arms to permit the cell to fit into the 5-inch wide sample compartment of the Perkin-Elmer UV-VIS spectrometer. Two 0.25-inch i.d. stainless steel tubes were silver-brazed onto opposing arms of the cell to permit purging of the cell with dry air (dew point < -75°C). The cell length was 5.6 inches for the infrared path, and 4.8 inches for the UV-VIS path. The clay-film holder consisted of either 3/8-inch or 1/2-inch i.d. stainless steel washers. The 3/8-inch size was the minimum that could readily accommodate the 1/8-inch by 1/4-inch (horizontal by vertical) rectangular UV-VIS beam.

The above cell was used to record the chemisorption of an aromatic compound onto a clay film as a function of humidity. A few tenths of a gram of a pure aromatic compound were added to the

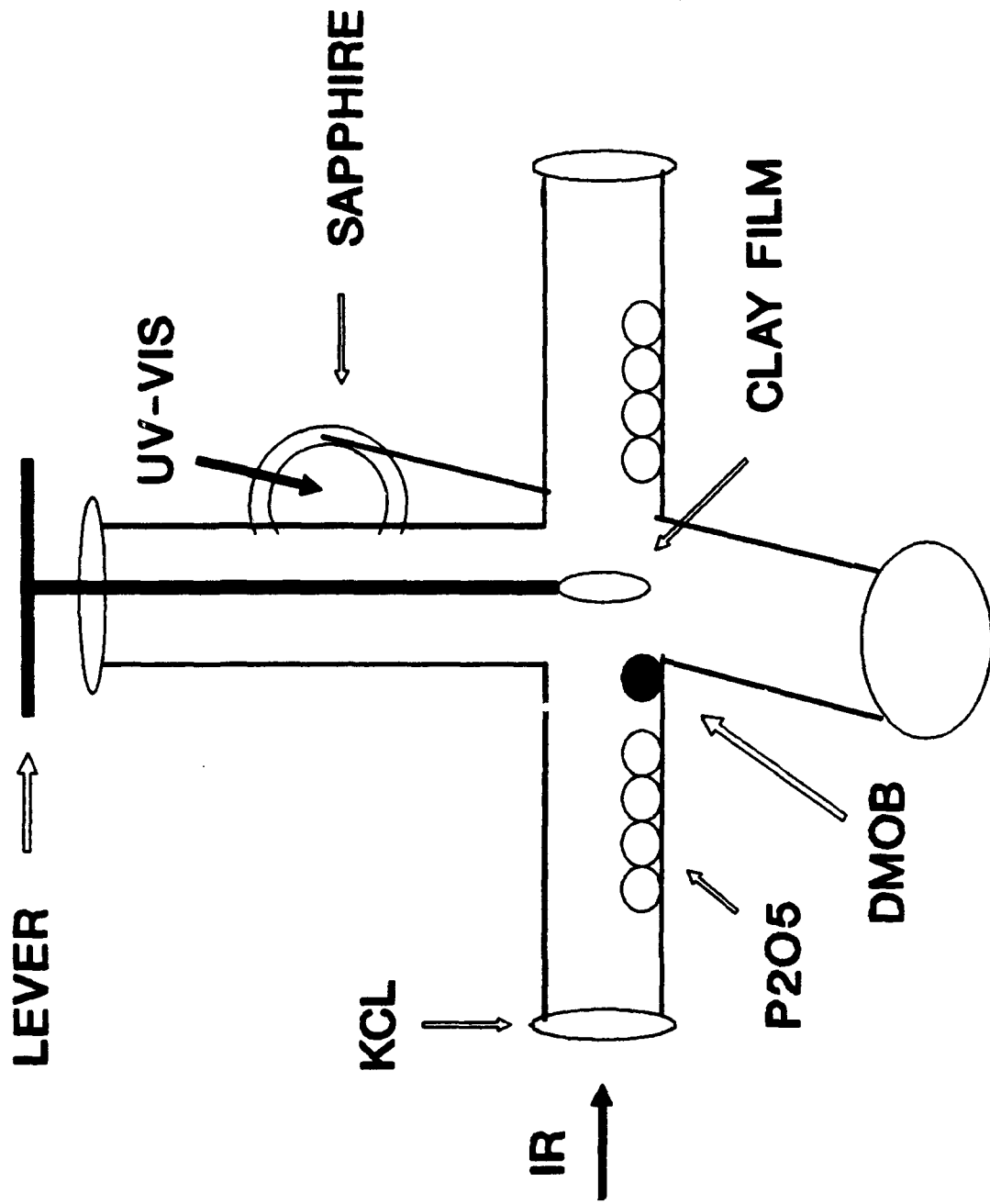


Figure 1. Clay-Film Cell for Tandem FT-IR and UV-VIS Experiments.

cell at the start of an experiment along with P_2O_5 desiccant. The aromatic compound was placed in the center of the cell, and the desiccant was placed immediately next to it in all four horizontal arms. This operation was carried out in a glove bag of dry air to protect the desiccant from atmospheric moisture. A clay film was introduced to the cell within the next several minutes. This caused the clay film to gradually dry over a period of several hours. The removal of water from the clay was typically accompanied by the chemisorption of the aromatic compound onto the clay from the vapor phase.

A clay film was formed by pipetting 1 mL of a 2 mg/mL clay suspension onto a polyethylene sheet. The area of the dried film was typically 2 cm², resulting in a 1 mg cm⁻² density. Since the infrared beam had a diameter of 0.2 cm, the mass of the clay that was exposed to the infrared beam was about 30 ng. The corresponding mass for the UV-VIS beam was about 200 ng.

D. CLAY EXTRACTION

1. Sample Preparation

An aliquot of the stock clay suspension obtained via the clay preparation procedure of Section II-A was placed in a mortar and heated until all of the water was vaporized. Typically, 0.5 grams of dehydrated clay was used for extractions. The clay was ground into a fine powder with a mortar and pestle, transferred to

a modified 100 mL round-bottom (RB) flask, and refluxed overnight (about 16 hr) in 70 mL of a liquid aromatic contaminant under a nitrogen atmosphere (Figure 2). The RB flask was outfitted with a gas inlet tube and Teflon® valve to permit the nitrogen to purge the apparatus prior to the start of refluxing. After this initial purge, the Teflon® valve was shut off, and the nitrogen inlet tube was transferred to the top of the condenser to minimize loss of the refluxing fluid.

After the refluxing was finished, the clay was separated from the solvent either by filtration or by decanting the RB flask and vaporizing the remaining liquid. The clay residue was then subjected to the Soxhlet extraction procedure given below.

2. Soxhlet Extraction

Soxhlet extraction is a method whereby a sample mixture is repeatedly exposed to a pure recycled solvent to enable even barely soluble components of the sample to be collected. A schematic diagram of a Soxhlet extractor is shown in Figure 3. The solvent is placed in a heated round-bottom (RB) flask, and a sample is placed in a cellulose thimble in the upper compartment. The solvent vaporizes from the RB flask, flows up to a condenser, and drips down into the upper compartment. When the upper compartment becomes full, the liquid siphons back into the RB flask. This process repeats indefinitely, causing extracted solute to accumulate in the RB flask.

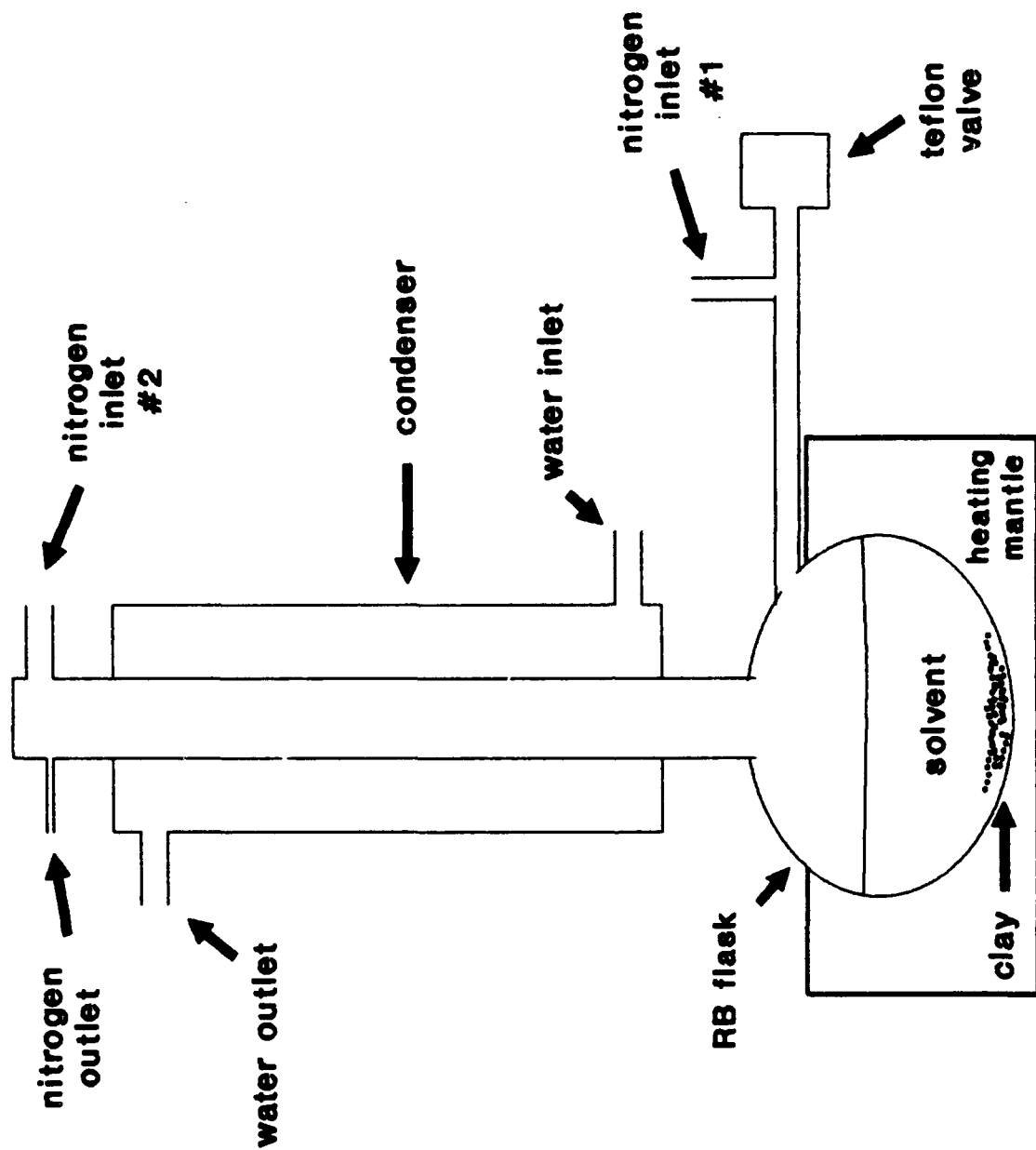


Figure 2. Apparatus for Refluxing Clay Under Nitrogen.

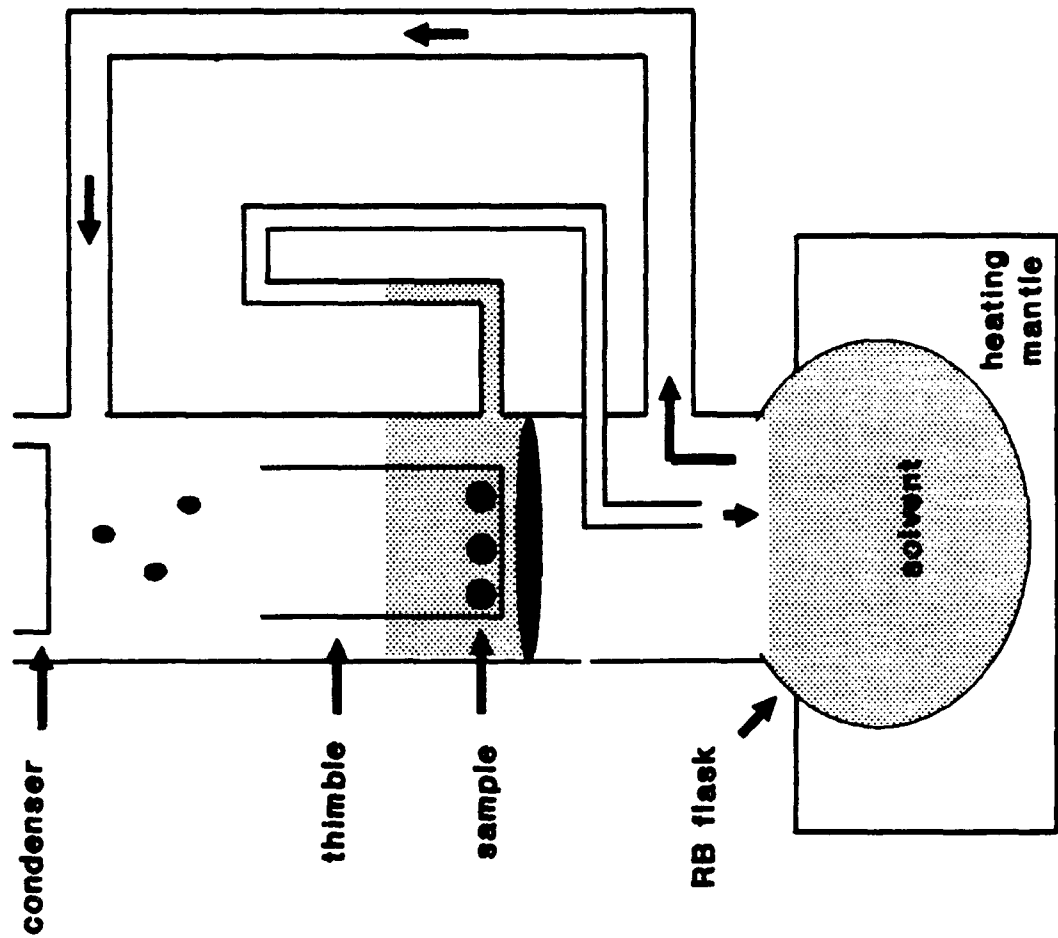


Figure 3. Soxhlet Extractor.

The above procedure was applied to numerous samples of montmorillonite powder containing adsorbed fuel components. Usually, several solvents had to be used on a given sample because no single solvent could extract all of the reaction products.

3. Supercritical Fluid Extraction

Supercritical fluid extraction (SFE) is a method whereby a highly compressed gas (typically 100 atm), whose temperature is too high to permit condensation, is used to extract components from a solid sample (Reference 4). SFE is often more effective than conventional extraction techniques because supercritical fluids can penetrate into microporous materials more readily than liquids can. Carbon dioxide is usually used as the solvent for SFE because of its general effectiveness and its relatively low supercritical temperature (31°C) and pressure (73 atm).

A simple SFE apparatus has been designed for testing the effectiveness of the SFE technique on contaminated clays (Figure 4). A pump is required in this system since commercial cylinders of carbon dioxide are not pressurized as high as the 73 atm minimum pressure required to produce a supercritical fluid state. The sample container and gas transfer tubes must be heated with an oven or heating tape (not shown) to achieve the requisite 31°C temperature. The sample container is a stainless steel cylinder with a lid. Bolts are used to press the lid down on an o-ring to produce a high-pressure seal.

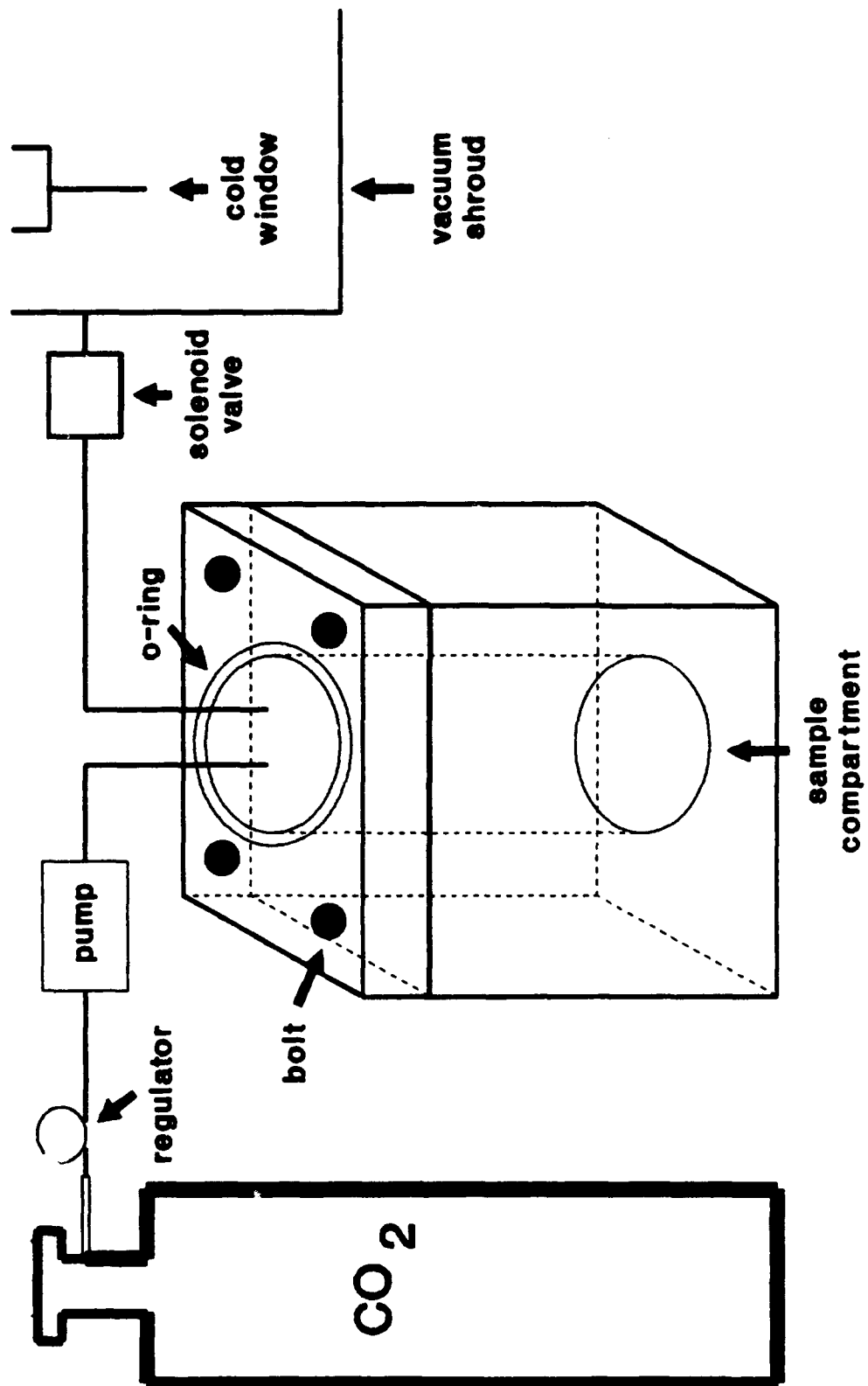


Figure 4. Apparatus for Matrix Isolation of Supercritical Fluid Extracts.

The effluent from the above apparatus is condensed onto a cryogenically cooled mirror to isolate the extract in a carbon dioxide lattice. A pulsed solenoid valve limits the rate at which the effluent emerges from the outlet. This scheme does not enable the extracted material to be separated into its component substances but it will determine whether SFE is effective in removing sorbed substances from contaminated clays.

4. Sample Heater for Matrix Isolation

A sample heater was constructed for the MI apparatus to: (1) test whether simple distillation of clay extracts would effectively separate component compounds; and (2) provide a means of obtaining reference MI spectra for compounds with low vapor pressures.

The heater (Figure 5) consisted of two 1000-ohm enamel wirewound power resistors connected in parallel to a variable transformer. A temperature of 335°C was observed when this heater was operated in ambient air at 136 volts. In normal use, the heater was interfaced with the shroud of the expander module of a closed-cycle refrigerator (Air Products Displex Model CSA-202), and was operated in a vacuum.

The sample material was placed at the end of a 5/16-inch o.d. stainless steel tube which fitted snugly inside of the hollow cores of the power resistors. The end of this tube contained a removable insert constructed from a 1/4-inch i.d. Swagelock®

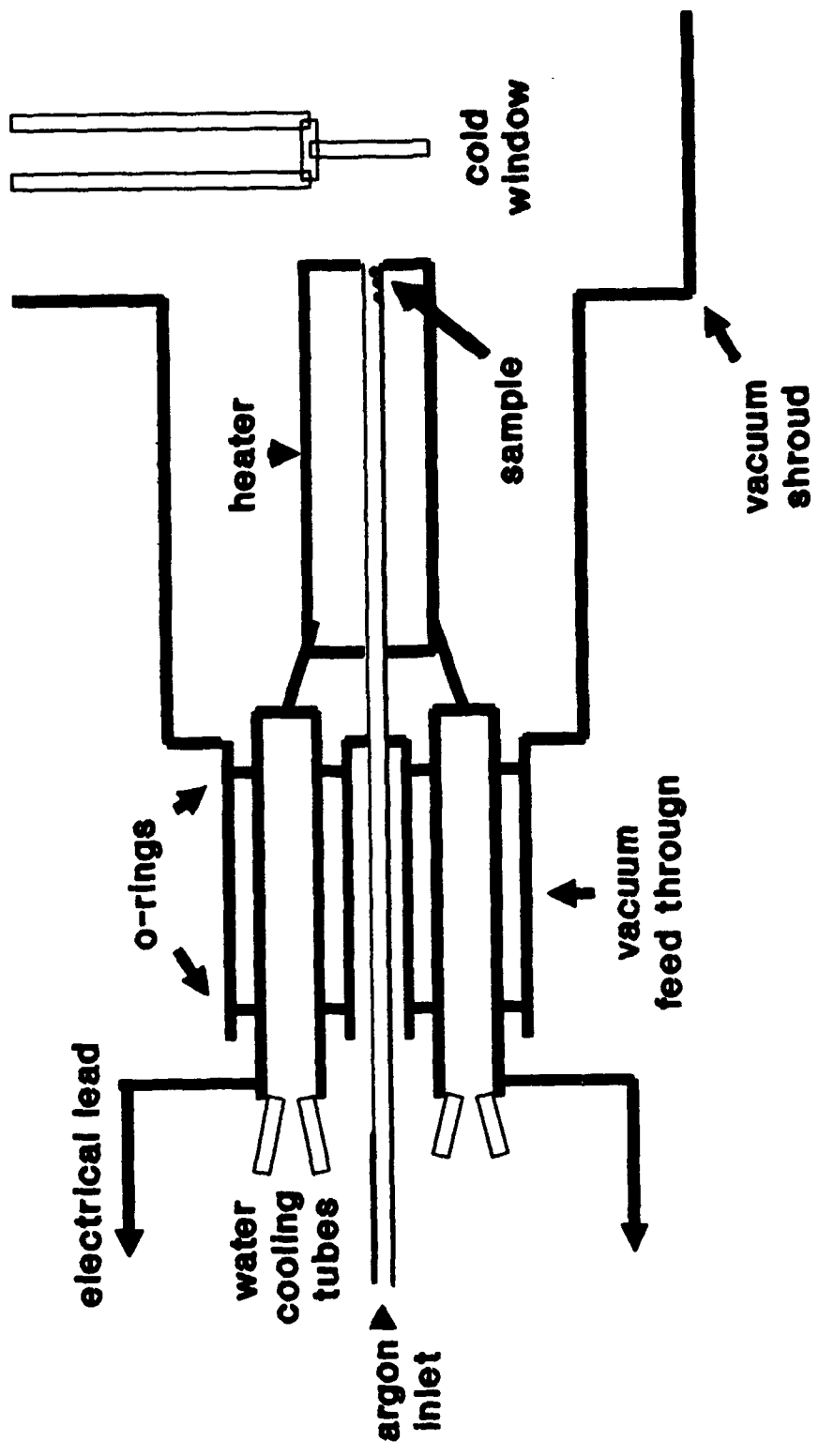


Figure 5. Sample Heater for Matrix-Isolation Experiments.

ferrule soldered to a circular piece of stainless steel foil with a pinhole in it. This insert allowed a sample to be easily placed into the core of the heater and prevented the sample from being sucked out of the tube during evacuation.

The heater was located within a cylindrical cooling jacket constructed from concentric 1-5/8-inch o.d. and 1-1/8-inch o.d. copper tubes, 4-1/2-inches long. One end of this jacket was silver-brazed to a 1-3/4-inch square stainless steel flange with a 1-inch entrance hole in it; the other end was silver-brazed to a flange through which the sample tube and two electrical feed-throughs passed. The feed-throughs consisted of concentric 1/4-inch and 3/8-inch o.d. stainless steel tubes separated by two pairs of #007 Viton® o-rings. Both tubes were filed on a lathe to create enough space for the o-rings to fit between them. Each o-ring was held in place by a pair of 1/4 inch i.d. Swagelock® ferrules which were silver-brazed into position and filed so that their outer diameters were smaller than the outer diameters of the o-rings. The electrical current passed through the 1/4-inch stainless steel tubes and through short pieces of copper wire to the power resistors. These tubes were cooled by transporting water in and out of them via two 1/8-inch o.d. stainless steel tubes. Originally, the feed-throughs were designed so that the cooling water passed between the 1/4-inch and 3/8-inch stainless steel tubes but this scheme was abandoned after failure of an o-ring enabled cooling water to leak directly into the evacuated shroud.

A thoroughly dry sample tube was essential to the effectiveness of the heater since a moist tube resulted in highly absorbing ice bands which obscured a large part of the most useful regions of the infrared spectrum. The tube was dried, while empty, by heating it under a vacuum at the maximum temperature to be used for the sample. The dryness was maintained during the insertion of the sample by flowing a dry vapor through the tube.

One deficiency of the above heater was that it contained exposed areas of relatively high voltage, and was thus vulnerable to catastrophic short circuits. This deficiency will be minimized by replacing the 1000-ohm power resistors with 2-ohm resistors as soon as the latter become available. The smaller resistors will be safer because they will be operated at much lower voltages than those presently required for the larger resistors.

5. Apparatus for: Gas Chromatography / Matrix Isolation / FT-IR Spectroscopy

The matrix-isolation (MI) apparatus used in conjunction with the current study on clays is largely the same as described in a previous technical report (Reference 5). A cryogenic refrigerator freezes a sample material within a lattice of an inert solvent such as argon. The sample molecules are thereby isolated at a density similar to that of a vapor, but without the rotational motion of the vapor state. The absence of rotational structure in

the infrared spectrum simplifies assignments of vibrational bands and facilitates identification of the absorbing specie.

Infrared reference spectra for pure compounds are obtained by spraying a gaseous sample/argon mixture onto an infrared transmission window. A similar procedure can be used for analyzing the effluent of a gas chromatograph (Reference 6). Such a scheme is useful for the identification of reaction products extracted from contaminated clay samples.

To implement the GC/MI/FTIR method, it is necessary to calculate the sensitivity needed to detect solutes eluted from a gas chromatograph with an infrared detector. The quantity of solution used for GC analysis is typically 1 microliter, or roughly 1 mg. A typical quantity of sample needed to produce an optimum infrared spectrum is about 30 ng, assuming that the sample diameter and beam diameter are both about 2 mm. Thus a solute/solvent mass ratio of 1:30 should produce a strong solute spectrum, and a much less favorable ratio may still be adequate to enable a compound to be detected and identified.

The sample diameter limitation of 2 mm has significant consequences in the design of a GC/MI/FTIR experiment. First, the tip of the GC column must be placed roughly 1 mm from the substrate. Transmission windows are unsuitable under these circumstances because they have insufficient heat conductivity to freeze the argon carrier gas at the required rate. Therefore, a gold-coated copper mirror (M1 in Figure 6) was used instead. Additional mirrors were used in conjunction with the copper mirror

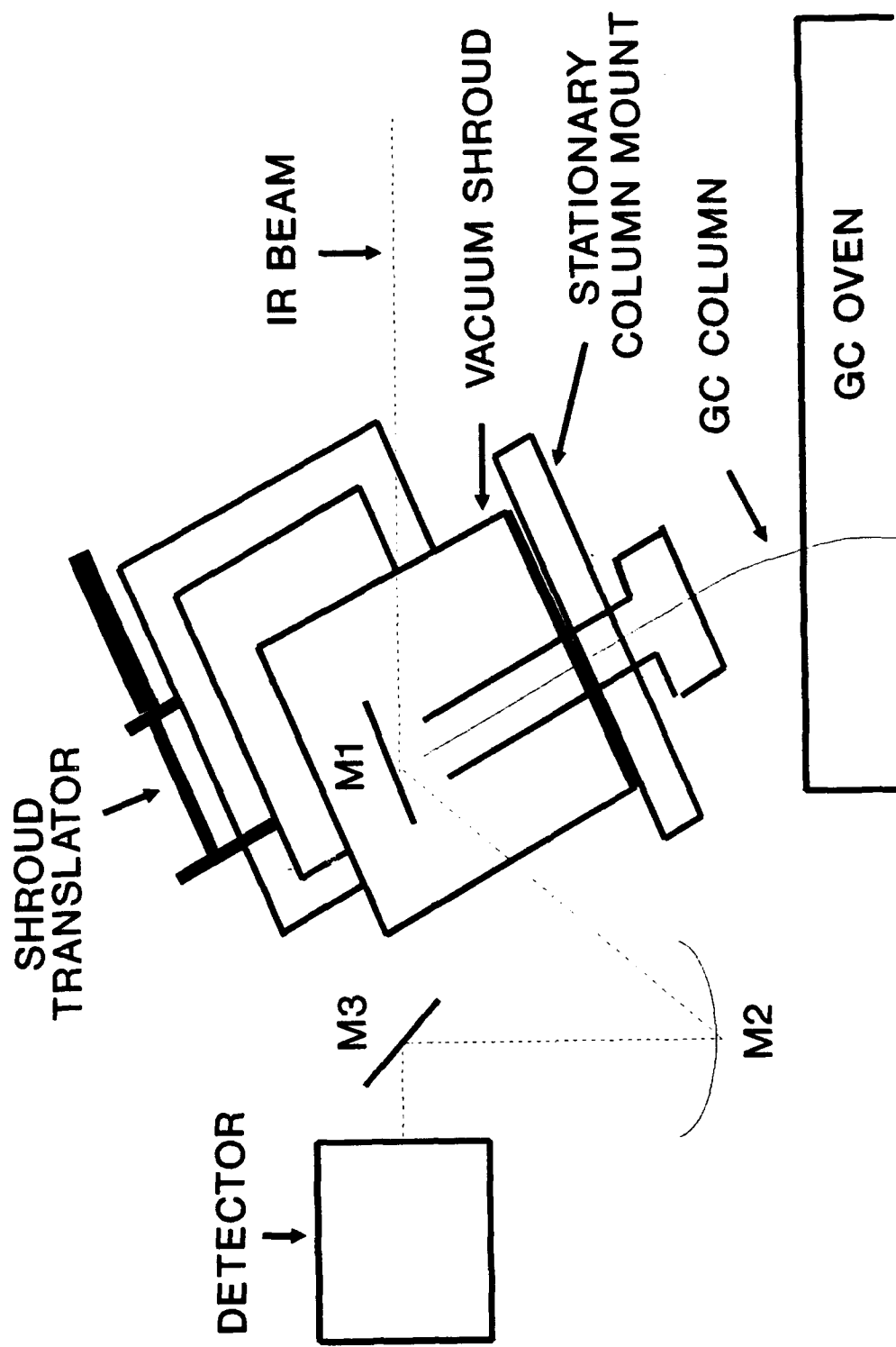


Figure 6. GC/MI/FTIR Apparatus.

to steer the infrared beam to the detector compartment of the FT-IR spectrometer and to maintain an optimum focus on the detector. Mirror M2 of Figure 6 was placed at approximately one focal length from the copper mirror to collimate the reflected infrared beam. Ideally, this mirror should be an off-axis parabola but a spherical mirror was found to be a satisfactory substitute. The collimated infrared rays were later focused onto an MCT-A detector using the focusing mirror supplied with the FT-IR instrument.

A second consequence of the FT-IR sample size limitation is that the argon carrier gas must be supplemented with a large excess of helium (which is noncondensable) to avoid producing a matrix which is too voluminous to fit between the end of the GC column and the surface of the copper mirror. It has been found (Reference 6) that a mixture of 2 percent of argon in helium is optimum under these circumstances. Alternatively, pure helium carrier gas can be used to obtain spectra of pure condensed samples (i.e. not matrix isolated). The latter option enables use of extensive commercial spectral libraries for compound identification. A library of matrix-isolation spectra is available but is too small to be of much practical use. The main advantage of matrix isolation in GC/MI/FTIR analysis lies in its ability to distinguish between molecular isomers.

A basic requirement of any GC analysis is that the output of the GC must be measured as a function of time. The simplest way to achieve this for GC/MI/FTIR is to record spectra continuously as the GC column effluent deposits onto a stationary mirror.

However, this method has two disadvantages: (1) the spectra are noisy because only a very limited amount of signal averaging can be done for any given effluent component; and (2) the sample may become totally absorbing in certain spectral regions because of accumulation of signal from many different molecular species. These problems can be alleviated by continuously moving the sample mirror as the effluent emerges from the GC column. Different species are thereby physically separated from each other on the mirror surface, and unlimited signal averaging can be done after all of the effluent has been deposited on the mirror. Figure 6 shows the apparatus that has been designed to implement this scheme. An optical translator moves the copper mirror in a plane which is parallel to the mirror surface, leaving the optical path of the infrared beam undisturbed. The GC column is held stationary during this motion by attaching it to a rigid plate that slides across an o-ring on the vacuum shroud. The optical translator can be controlled by using a computer-operated motor available commercially. The sample mirror can be accurately repositioned by the computer to any desired part of the chromatogram after the sample has been collected. Although only an x translator is shown in Figure 6, an xy translator can be substituted to make maximum use of the mirror surface.

A final requirement of the GC/MI/FTIR apparatus is that a heated interface must be inserted between the GC and MI components to accommodate the GC column. This has been done by inserting the end of the GC column through a 1/8-inch o.d. copper

tube and wrapping the tube with insulated 25-gauge nichrome wire. The nichrome leads have been connected to a variable transformer to enable heating of the interface to 200°C. The copper tube enters the shroud through a 1/8-inch Ultra-torr® stub containing a Kalrez® o-ring. The nichrome heater wire extends up to this Ultra-torr®; the final portion of the tube is short enough (about 1 inch) so that only a small drop in temperature occurs near the tip of the GC tube. Polyimide resin (Alltech Associates) provides a vacuum seal for the end of the copper tube.

SECTION III
RESULTS AND DISCUSSION

A. TANDEM FT-IR AND UV-VISIBLE EXPERIMENTS ON CONTAMINATED CLAYS

The reactivity of aromatic compounds on many types of clays results from the presence of mobile metal ions within the stacked-plate structure of the clays (Reference 7). This property has enabled clays to be used as catalysts for reactions involving aromatic organic molecules. Montmorillonite clay is particularly effective as a reaction medium because of its unusually high specific area of about $500 \text{ m}^2\text{g}^{-1}$. Each layer of montmorillonite consists of an aluminosilicate structure (approximate formula of $\text{Cu}_{0.25}\text{Al}_{1.5}\text{Mg}_{0.5}\text{Si}_4\text{O}_{10}(\text{OH})_2$ for the Cu-exchanged form) in which an octahedral Mg-Al layer is sandwiched between two tetrahedral silicon layers.

An FT-IR spectrum of Cu-exchanged montmorillonite is shown in Figure 7. The most prominent feature of this spectrum is the SiO stretching vibration at 1040 cm^{-1} (Reference 8). Al-Al-OH and Al-Mg-OH bending vibrations occur at 920 cm^{-1} and 840 cm^{-1} , respectively. Hydroxyl (OH) groups in the clay lattice structure, as well as water molecules in the clay interlayers are responsible for other prominent vibrational bands. The latter produces a broad band ($3600 - 2800 \text{ cm}^{-1}$) in the OH stretching region while the former produces a relatively sharp band at 3620 cm^{-1} . The corresponding bands in the OH bending region are not readily distinguishable.

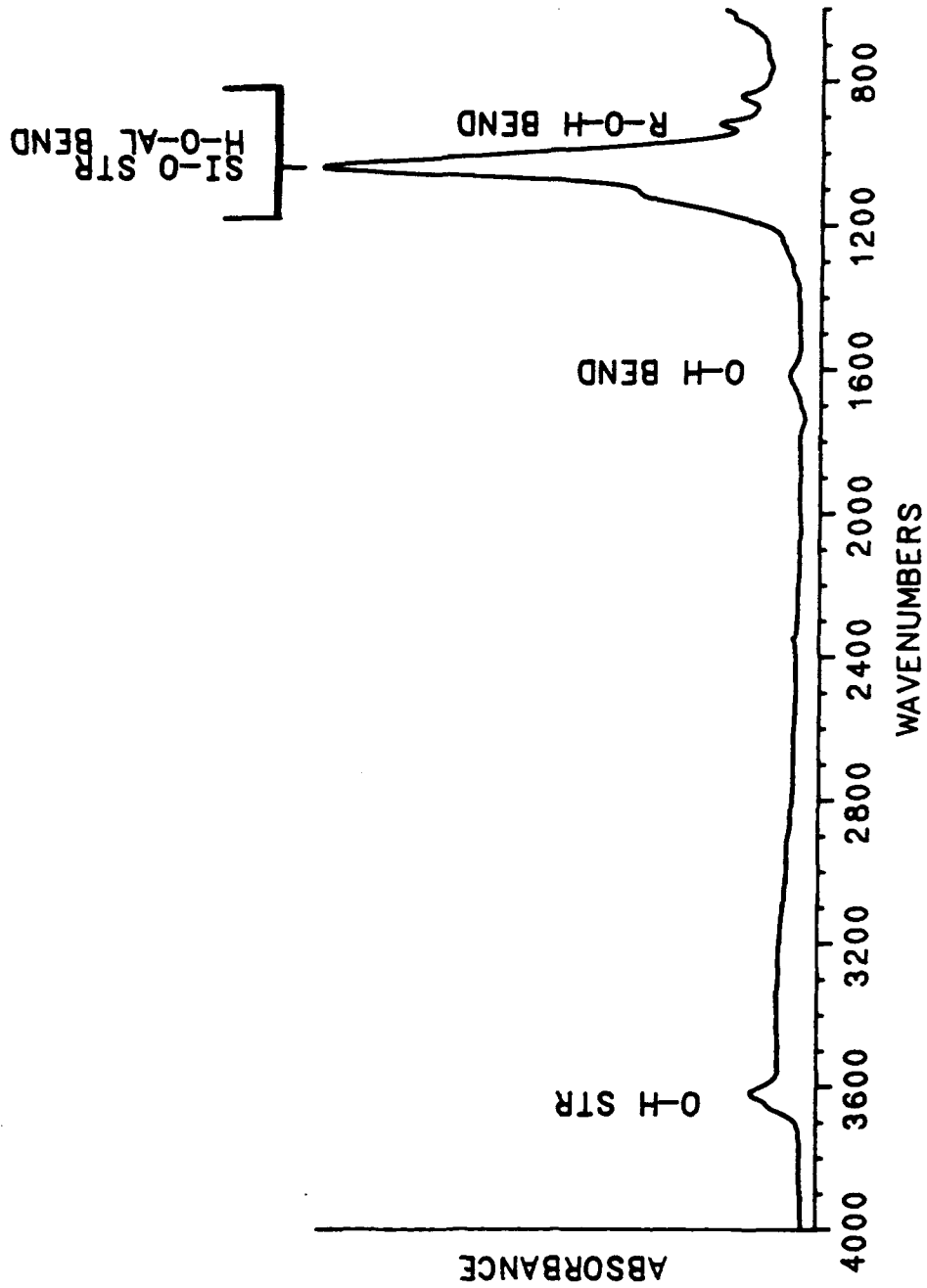


Figure 7. FT-IR Spectrum of Cu-Exchanged Montmorillonite.

The substitution of divalent Mg for trivalent Al in the octahedral layer of montmorillonite results in an excess of negative charge in the interlayers. This charge imbalance induces other positively charged atoms to intercalate into the interlayers of the clay, resulting in a highly acidic substrate. Ions commonly found in the interlayers of natural clays include Na^+ , Ca^{++} , Cu^{++} , and Fe^{+++} . Under typical natural conditions, these ions are surrounded by water molecules and the clay is inert to most aromatic molecules. However, under warm and/or dry conditions, many aromatic molecules have been found to undergo a charge transfer with transition metal ions such as Cu^{++} and Fe^{+++} . A schematic diagram of this charge transfer is shown in Figure 8 for benzene in Cu-montmorillonite.

The first system selected for the tandem FT-IR/UV-VIS experiments was p-dimethoxybenzene (DMOB) on Cu-montmorillonite (CUM). A brief discussion of this system is given below. A more thorough discussion has been presented elsewhere (Reference 9).

Figure 9 shows the growth of infrared bands on CUM as a function of the time of exposure to DMOB in the presence of P_2O_5 desiccant. For this experiment, the clay film was supported on a polyethylene substrate since it was too thin to be self-supporting. A remnant of a subtracted polyethylene absorption band can be seen at 1470 cm^{-1} . Vapor-phase DMOB did not have to be subtracted since it was too weak to be observable. The disproportionate growth of the bands in this spectrum indicates that at least two species are present. One of these species can be identified as physisorbed

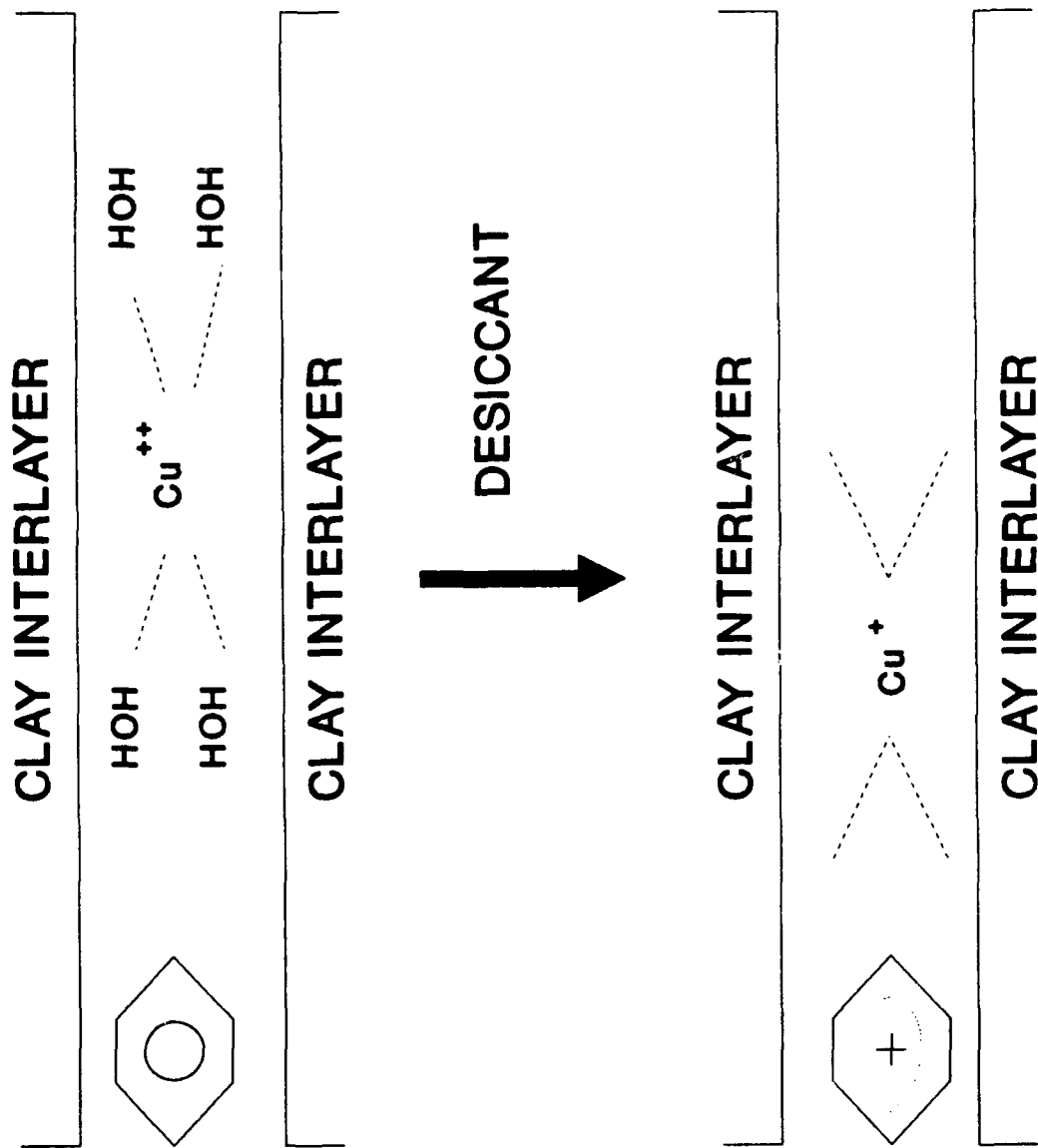


Figure 8. Charge-Transfer Reaction for Benzene on Cu-Montmorillonite.

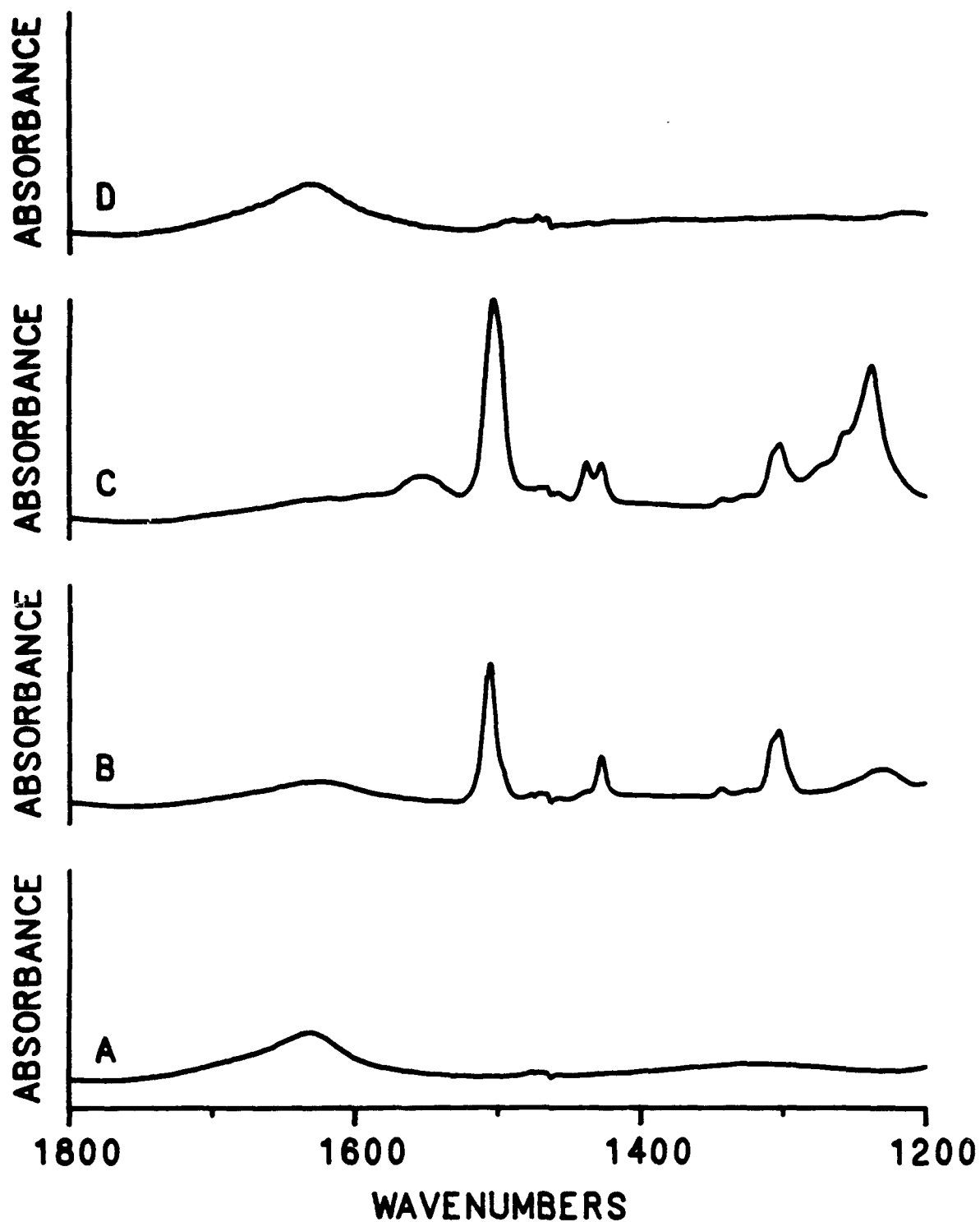


Figure 9. FT-IR Spectra of Cu-Montmorillonite + DMOB + P_2O_5 : A) at 0.0 Hr with Only Clay Present, B) at 3.0 Hr, C) at 69 Hr, D) After Reintroduction of Ambient Air and Removal of DMOB.

DMOB by comparison with a reference spectrum of a pure solid DMOB sample. Another can be identified as the cation of DMOB by comparing the UV-VIS spectrum of the clay sample with that of the spectrum of the DMOB cation that has been published in the manual of Shida (Reference 10).

A comparison of UV-VIS data of DMOB with corresponding FT-IR data is given in Figure 10. Part of the infrared spectrum was deleted from this figure because it contained strong absorption from the polyethylene substrate. The growth of the infrared bands in the 1525 cm^{-1} to 1200 cm^{-1} region is in qualitative agreement with the growth of the DMOB cation bands at 440 nm and 460 nm. Unfortunately, the UV-VIS bands were too strongly absorbing to permit a more accurate comparison.

The DMOB/CUM system is particularly simple because no reaction takes place other than a transfer of charge from the Cu^{++} ions on the clay to the aromatic molecules. Examinations of more complicated systems such as benzene/CUM and toluene/CUM are planned.

The spectroscopic absorption of an aromatic compound on a clay can be obtained as a function of adsorbed mass by interfacing a clay-film cell to a microbalance. Such an apparatus is in use at the University of Florida and is described in the literature (Reference 9). A similar cell is planned for use at Tyndall AFB.

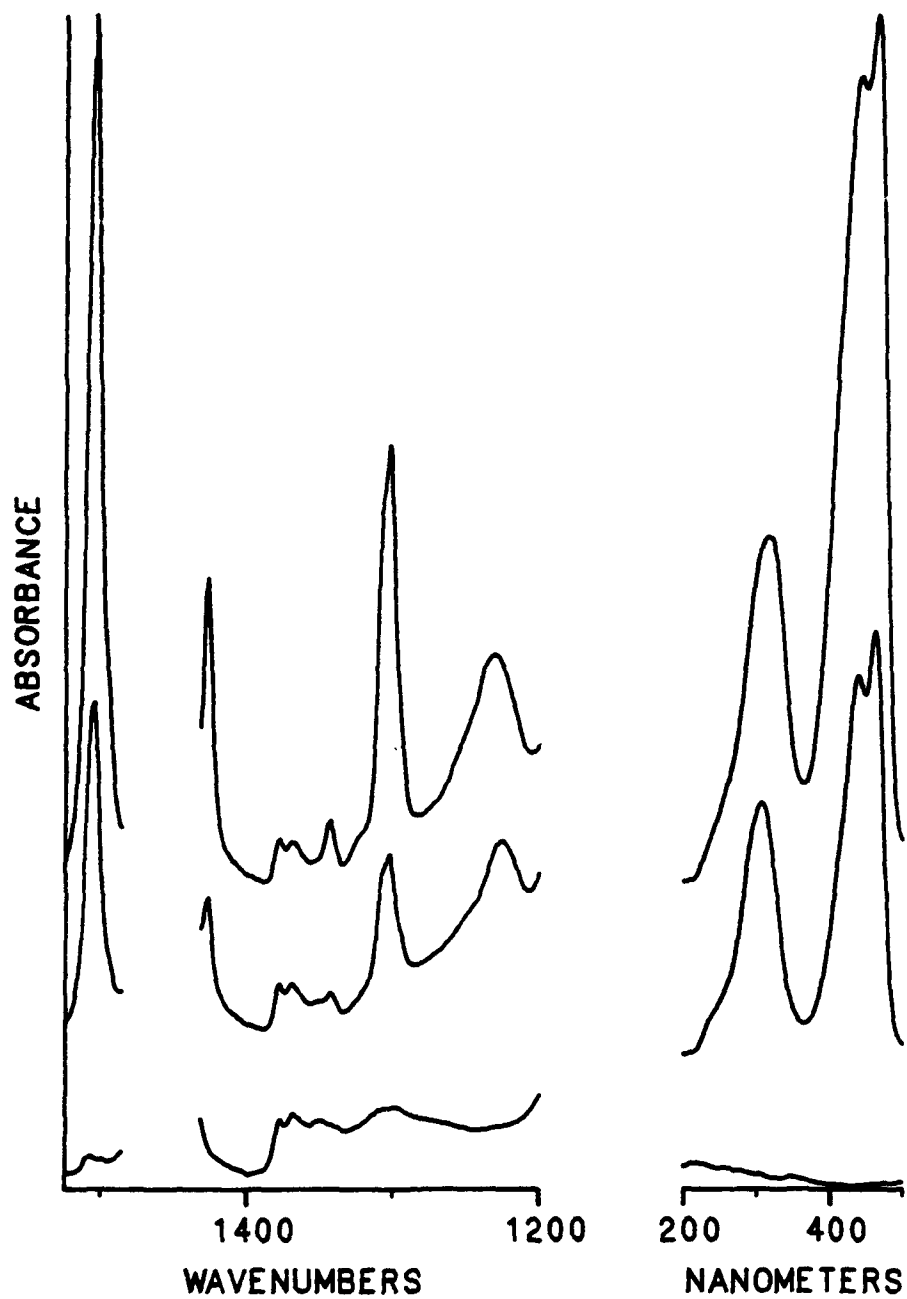


Figure 10. Growth of FT-IR and UV-VIS Bands of a DMOB/CUM Film as a Function of Elapsed Time of Dehydration: A) 0.1 Hr, B) 2.0 Hr, and C) 3.0 Hr.

B. ANALYSIS OF CLAY EXTRACTS

Examples of data collected during a typical clay extraction are shown in Figures 11 and 12. Figure 11 compares the FT-IR spectrum of a toluene-contaminated CUM sample with the corresponding spectrum of the acetone extract of this sample. The infrared absorption of the contaminant mixture is much stronger in the extract than in the original sample because the contaminant is much more concentrated in the former case. The relative intensities of absorption bands in the two spectra are not the same because the acetone extracted the components of the contaminant mixture with varying degrees of efficiency.

A GC/MS analysis of the extract yielded the chromatogram shown in Figure 12. This chromatogram has not yet been fully analyzed but the strongest peak has been identified as 1-methyl-2-phenylmethylbenzene, which is a toluene dimer. This result is in accordance with previous reports of multimer formation for some aromatic compounds adsorbed on CUM (References 11 and 12).

C. MATRIX-ISOLATION EXPERIMENTS WITH HEATED SAMPLES

The sample heater described in Section II-D was ineffective in separating the components of clay extracts because the differences between the vapor pressures of the components was not sufficiently diverse. However, the heater appears to be useful

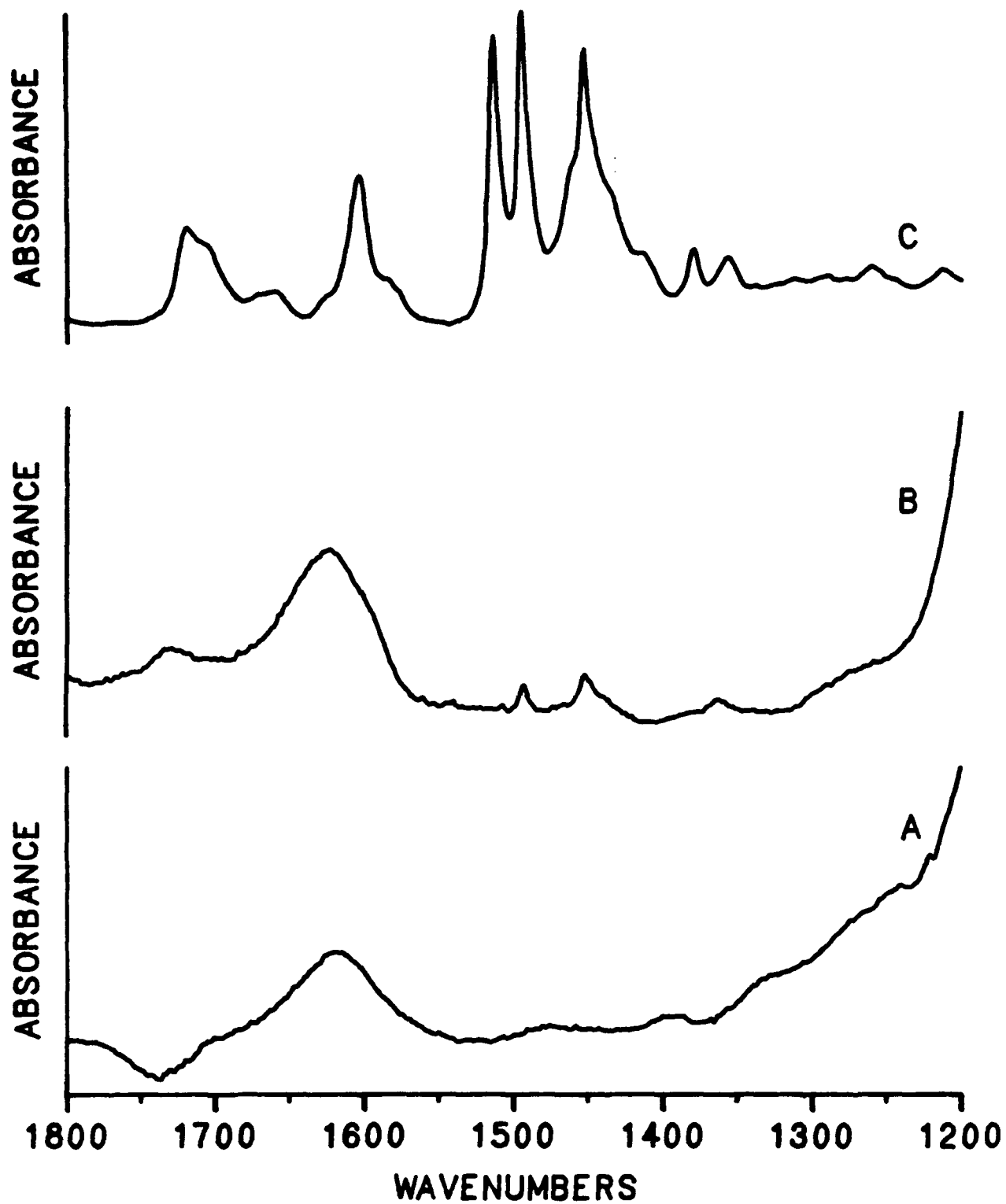


Figure 11. FT-IR Spectra of: A) Cu-montmorillonite in Ambient Air, B) Cu-montmorillonite After Refluxing in Toluene, and C) An Acetone Extract of Toluene-Contaminated Cu-Montmorillonite.

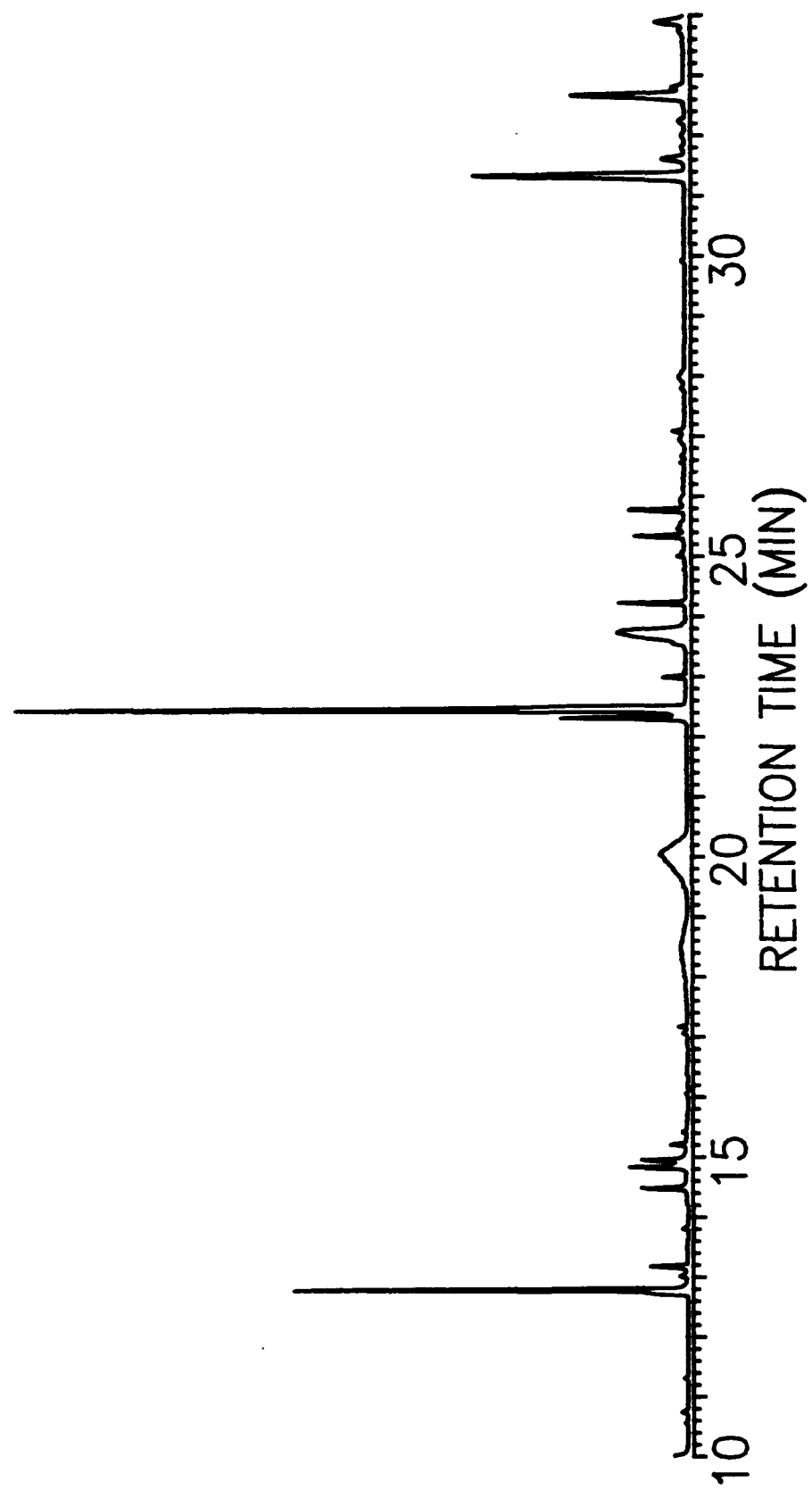


Figure 12. GC/MS Chromatogram for Acetone Extract of Toluene/CUM.

for obtaining reference spectra of pure compounds of low volatility.

SECTION IV

CONCLUSIONS

Two accomplishments have been made toward understanding the behavior of aromatic compounds on soils at the molecular level:

1. The analysis of a simple system: DMOB/CUM. For this system, reaction occurs only under very dry conditions, and proceeds only to the point where a charge transfer takes place between the Cu^{++} ions and the DMOB to form Cu^+ and DMOB^+ . As soon as water is reintroduced, the DMOB^+ cations revert back to neutral DMOB. This behavior seems to be characteristic of para-substituted benzene compounds (References 3, 9, 13, and 14).

2. The implementation of methods for extracting reaction products from clays and analyzing the extracts. The analysis of complex reactions on clays would be prohibitively difficult without the capability of separating components of reaction-product mixtures. Soxhlet extractions have been applied with some success and may soon be supplemented with SFE. GC/MS has proven effective in identifying some components of clay extracts and will shortly be complemented with the GC/MI/FTIR technique. These methods will be useful in analyzing systems such as toluene/CUM and benzene/CUM, where formation of multimers (i.e. dimers, trimers, ...polymers) has been reported (References 2, 11, and 12).

SECTION V
FUTURE PLANS

One near-term objective is to combine the SFE and GC/MI/FTIR techniques with traditional methods to analyze clay/aromatic systems in which a multitude of reaction products is formed. A second is to combine gravimetric measurements with FT-IR spectroscopy to relate a bulk property of a clay (grams of contaminant sorbed) to a molecular-level property (spectroscopic absorption).

Once an understanding of a few representative clay/aromatic systems has been achieved, it should be possible to predict the behavior of similar systems. The complexity of the systems will then be increased by replacing the clays with natural soil samples, and by using multicomponent fuels instead of individual aromatic compounds. A contaminant-transport model generated from these studies should be superior to empirical models based on bulk measurements since the latter may be based on false assumptions of molecular-level interactions and cannot be extrapolated with confidence.

One possible by-product of studies on contaminated clays might be the development of an artificially modified clay that efficiently traps any aromatic fuel molecules that sorb onto it. Such a clay could be used to contain a fuel spill within a localized area. One way of producing such a clay is to introduce

surfactants. Much research is already underway in this area (e.g. see Reference 15).

REFERENCES

1. Westall, J. and Hohl, H. "A Comparison of Electrostatic Models for the Oxide/Solution Interface," Advan. Colloid Interface Sci. **12**, 265-294 (1980).
2. Soma, Y., Soma, M. and Harada, I. "The Reaction of Aromatic Molecules in the Interlayer of Transition-Metal Ion-Exchanged Montmorillonite Studied by Resonance Raman Spectroscopy. 1. Benzene and p-Phenylenes," J. Phys. Chem. **88**, 3034-3038 (1984).
3. Soma, Y., Soma, M. and Harada, I. "Reactions of Aromatic Molecules in the Interlayer of Transition-Metal Ion-Exchanged Montmorillonite Studied by Resonance Raman Spectroscopy. 2. Monosubstituted Benzenes and Disubstituted Biphenyls," J. Phys. Chem. **89**, 738-742 (1985).
4. McHugh, M.A. and Krukonis, V.J. Supercritical Fluid Extraction: Principles and Practice, Butterworth Publishers (1986).
5. Johnston, C.T., Person, W.B., Vala, M.T., Tipton, T.L. and Stone, D.A. Spectroscopic Studies of Metal-Ligand-Surface Interactions, Final Report, AFESC, Tyndall AFB, Florida, ESL-TR-88-08 (1986).

6. Reedy, T.R., Ettinger, D.G., Schneider, J.F. and Bourne, S. "High-Resolution Gas Chromatography / Matrix-Isolation Infrared Spectroscopy," Anal. Chem. 57, 1602-1609 (1985).
7. Laszlo, P. "Chemical Reactions on Clays," Science 235, 1473-1477 (1987).
8. Borchardt, G.A. in Dixon, J.B. and Weed, S.B., (Eds.), Minerals in Soil Environments, Soil Science Society of America (1977), Chap. 9.
9. Johnston, C.T., Tipton, T., Stone, D.A., Erickson, C. and Trabue, S.L. "Vibrational and Electronic Spectroscopic Study of p-Dimethoxybenzene," Langmuir, to be published.
10. Shida, T., Electronic Absorption Spectra of Radical Ions, p. 257, Elsevier, Amsterdam (1988).
11. Pinnavaia, T.J., Hall, P.L., Cady, S.S. and Mortland, M.M. "Aromatic Radical Cation Formation on the Intracrystal Surfaces of Transition Layer Lattice Silicates," J. Phys. Chem. 78, 994-999 (1974).

12. Mortland, M.M. and Halloran, L.J. "Polymerization of Aromatic Molecules on Smectite," Soil Sci. Soc. of Amer. J. 40, 367-370 (1976).
13. Pinnavaia, T.J. and Mortland, M.M. "Interlamellar Metal Complexes on Layer Silicates. I. Copper(II)-Arene Complexes on Montmorillonite," J. Phys. Chem. 75, 3957-3962 (1971).
14. Soma, Y., Soma, M. and Harada, I. "Raman Spectroscopic Evidence of Formation of p-Dimethoxybenzene Cation on Cu- and Ru-Montmorillonites," Chem. Phys. Lett. 94, 475-478 (1983).
15. Hayes, K.F. "An FT-IR Spectroscopic Investigation of Surfactant Adsorption at the Mineral-Water Interface," 1989 USAF-UES Summer Faculty Research Program, AFESC, Tyndall AFB, Florida.

APPENDIX A

PROGRAM DESCRIPTIONS AND SOFTWARE CODES FOR THE WORK STATION OF THE NICOLET MODEL 740 FT-IR SPECTROMETER

Appendix A contains:

1. An alphabetic listing of user-defined macrocommands executed from the FT-IR SX operating system.
2. A list of FORTRAN programs and commands that are executed from the NICOS operating system.
3. An alphabetic listing of macro software codes.
4. An alphabetic listing of FORTRAN software codes.

Each macro command given below is followed by :

1. A list of submacros, if any.
2. A list of FORTRAN programs and NICOS data files used by the macro.
3. A brief description of the macro.

The names of all of the listed macros obey the following rules:

1. They consist of three alphanumeric characters.
2. They identify the macro as a parent macro or a submacro depending on whether the third character is a letter or number, respectively.
3. Wherever possible, they alphabetically group the parent macros with their associated submacros. Exceptions occur when a submacro belongs to more than one parent macro, and when the submacro is a conditional macro.

ABL Blanks absorbance file DFN. It is used for clearing out a file prior to sending FORTRAN output to it.

AOB SB1, SBFTYP.FCP
Converts a single-beam file (RFN) to "auto-absorbance" (DFN) using the maximum intensity between XSP and XEP as the background for RFN.

ALN M17, M19, SA0; TMXFTR.GCP
Aligns the "fixed" mirror of the interferometer. It is essentially the same as the system macro of the same name except that it allows the detector, resolution, and aperture to be changed without running a separate macro.

ASC Compresses a block of scratch files into an ASCII file and directs the output through the printer port to a logging

file on a second computer. If no output is obtained, toggle the DCE/DTE switch next to the printer port. If strange characters are produced, set the receiving computer to "parity = none". If some of the data gets lost, reduce the baud rate of the Nicolet computer. The program can be tested by sending the output to a Nicolet printer instead of a second computer.

- ATG** AT1, AT2. AUTIT.FCP, AUTIT1.FCP
This is a documentation macro only. It describes how to generate automated titles in the course of executing an automated data-collection macro such as SAM. When this procedure is implemented, file titles can be generated which include automatically incremented file extensions, collection dates, and elapsed times. Since the file title is restricted to 40 characters, it may not always be possible to select all of the possible options that are available.
- BAK** SA0, M17, M23
Collects and displays a background spectrum. The interferogram is copied to Nicos file I.EXT[,IRDATA] so that the raw data can later be saved on floppy disk, if desired. The single-beam spectrum is placed in file BFN.
- BCR** BC1
Corrects a spectrum baseline interactively. It divides the spectrum into several segments and prompts the user to correct these segments individually.
- BLX** BLANKX.FCP, FTPARM.FCP
Blanks several wavenumber ranges of a spectrum. It is useful for plotting a spectrum in which the absorbance is off scale in some wavenumber ranges.
- BLZ** BLANKZ.FCP, BLANK.DAT
Sets the absorbance to zero over selected wavenumber ranges of an absorbance file, and uses baseline adjustment to avoid sudden offsets in the y-axis at the ends of the wavenumber ranges. It is useful for extracting the spectrum of one compound from those of other compounds present in the same sample.
- BSL** ABL, BS0-BS2, M20-M21; BASLIN.DAT, BASLIN.FCP, FTPARM.FCP, MATINV.FCP, SPLINE.FCP
Corrects the baseline of an absorbance spectrum automatically. It is intended for eliminating fringes from matrix-isolation spectra. It resets the absorbance to zero at the minimum absorbance point in every equally sized block of words in a spectrum; and applies cubic spline interpolation to generate a baseline for all other points. The baseline points are sent to BASLIN.DAT and to scratch file 0 so that badly chosen points (e.g. those that are within a molecular absorption band) can be deleted and the

baseline recomputed. Subtraction of water and carbon dioxide absorption should be performed (using SVI or SVP) prior to running this program. This program is unsuitable for spectra having many wide absorption bands.

- DCP Scales several absorbance spectra so that a specified reference peak appears to have the same absorbance on the display screen for all of the spectra. The scaling affects only the screen display, and does not alter any stored files.
- DER RESOLU.FTN
Deresolves an absorbance file to 16 wavenumbers resolution for library searches, using software supplied by Nicolet.
- DSC DS1
Compares successive fragments of two files (RFN and DFN) on the display screen using constant absorbance per inch. The minimum absorbance is obtained separately for RFN and DFN by autoscaling. The maximum absorbance is obtained for RFN by autoscaling, and for DFN by adding the range of RFN to the minimum absorbance of DFN. This scheme enables comparison of spectra having markedly different baselines.
- DUM This is a dummy macro used to document all other macros.
- EMP M99; EMPFIL.FCP
Copies a spectral file into all vacant files in a block of scratch files. This prevents an "invalid fsb" message from occurring during execution of the TTL macro.
- FSP FSPRN.FCP
Lists a sector of scratch file data on the display screen.
- ITG INTGRL.FCP, INTSEG.DAT
Integrates selected wavenumber ranges of an absorbance file and sends the output to the printer. These ranges must first be entered into Nicos file "INTSEG.DAT" using the formatting described in INTGRL.FOR.
- LBI LB1, LB2
Moves a block of I.EXT[,IRDATA] interferogram files created by macros "BAK", "STI", or "SAM" to a user-specified location after initially transferring them to a block of scratch files. It is important to select the intermediate scratch files so that no important spectra are overwritten.
- LBM LB1, LB2
Copies a block of scratch files to a user-specified disk location.
- LTR LETTERS.FCP, GPLOT.LIB[HLIB,ZETA8]
Sends text from the keyboard to the plotter. The size and orientation of the letters are user-specified. This macro

is needed for adding text to rotated plots created by macros PLI and PLC since these plots are oriented differently than the rotated plots produced by Nicolet's software.

- NSF Copies a block of files from Nicos to the scratch area.
- NXP Advances the page of the Zeta-8 plotter.
- PCP PC1
Performs the same function as macro DCP except that the output for PCP goes to the plotter.
- PFB BENPAR.740
Reads parameter file PFN of BENPAR.740, and prints pre-selected parameters.
- PLC PL1-PL4, PL7, M30-M32, M40-M49; AXIS2.FCP, FSBTIT.FCP, GPLOT.LIB[HLIB,ZETA8], PLOTFT.FCP, PLT.DAT, PLPARR.FCP, PLPARW.FCP
Compares several plots on one x-axis. It enables the user to control plot parameters to the fullest extent permitted by standard FORTRAN plotting subroutines. Many of the standard FTIR parameters are redefined within this program; the initial settings of these parameters are therefore saved at the beginning of this program in PFN=49, and retrieved after the plotting is over. All of the plotting parameters are read from macros rather than from FORTRAN subroutines so that the user can reorganize the setup of the program without editing FORTRAN programs. For example, the user can easily change a defaulted parameter by inserting an extra line of keyboard input. Plotting parameters are saved in NICOS file PLT.DAT each time PLC is executed so that the program can be rerun with these parameters without having to reenter them.
- PLI PL1-PL6, M40-M46, M48-M49; AXIS2.FCP, FSBTIT.FCP, GPLOT.LIB[HLIB,ZETA8], PLOTFT.FCP, PLT.DAT, PLPARR.FCP, PLPARW.FCP.
Plots N files on N x-axes. Otherwise, PLI uses the same algorithms as PLC.
- PMY Prints y-axis limits for a block of FTIR files using multiple screen displays.
- PPP PP1
Plots an FTIR file and prints wavenumbers of spectral peaks.
- RES RESOLU.FTN
Changes the resolution of an absorbance file using software supplied by Nicolet.
- R#W PFB

Sets the resolution by reading a parameter file. The "#" character of R#W is a code that designates a parameter file, and consequently, the resolution. The codes are set at 0, 1, 2, 3, or 4 for resolutions of 0.5, 1, 2, 0.3, and 4 wavenumbers, respectively. The R#W macros can be used in conjunction with the ASG command to enable the resolution to be set by pressing a single key.

- SAM** SA0, SA2-SA5, SA9, M12-M15, M17, M50-M52; AUTIT.FCP, AUTIT1.FCP
Collects, displays, and auto-titles a block of sample spectra. Delay times between spectra can be chosen to be automated or manual. Input can be read from a file of default parameters (M51 or M52) if desired. Two screen displays of the spectrum are presented so that both a full spectrum and a magnified segment of the spectrum can be obtained without using cursor keys.
- SBS** SB0-SB1; SBFTYP.FCP
Regenerates a single-beam file from an absorbance file (DFN) and a second single-beam file.
- SMH** SMOOTH.FTN
Smooths a spectrum using Nicolet's software.
- STI** SA0, SA2-SA5, SA9, M12-M15, M17, M23, M50-M52
Collects, displays, and manually titles a block of sample spectra. Except for titling, STI is identical to SAM.
- STK** STICK.DAT, STICK.FCP
Generates a stick spectrum from absorbance data in Nicos ASCII file STICK.DAT. See directions in STICK.FOR for setup of STICK.DAT.
- SUM** SU0; SUMSPC.FCP
Similar to Nicolet's ADD operation except that the output goes to a new scratch file, and a constant value can be added to a spectrum.
- SVI** SV0-SV3
Interactively subtracts individual water and carbon dioxide vapor bands from an absorbance spectrum using reference spectra from the [ROOT,VAPOR] subdirectory. The generic names of the reference spectra are listed within the SVI program. The file extensions are set to either 0 or 2 for resolutions of 0.5 and 2 wavenumbers, respectively, using a formula listed in submacro SV3. For other resolutions, it is necessary to create additional reference spectra (e.g. by using the RES macro) whose names are compatible with the SVI macro.
- SVP** SV0-SV3

Interactively subtracts water and carbon dioxide vapor bands using one multiplier for each of these two molecules. Otherwise, SVP is identical to SVI.

TTL Prints a block of FTIR file titles and starting times using multiple screen displays.

FORTRAN PROGRAMS RUN FROM NICOS

AMAC.FOR This program alphabetizes a macro directory by alphabetizing the ASCII file MACROSAVE created from MACTBL and MACDIR. MACROSAVE must first be generated via the following set of NICOS commands:

```
COPY MACTBL/MACTBL.SAV
COPY MACDIR/MACDIR.SAV
MACGET
@MACROSAVE
```

The macros comprising MACROSAVE are sent to a set of generic files called JUNK, and then transferred alphabetically to output file AMAC.NEW. After the program has ended, the intermediate JUNK files can be deleted by using the command:

```
DEL JUNK.*:D
```

To complete the replacement of the old macro directory with the alphabetized directory, it is necessary to apply the KIL command to the old macro directory and then read the AMAC.NEW file with the MACCRT command.

BAUD# (where # = 300, 600, 1200, 2400, 4800, 9600) These programs reset the baud rate of the printer port. They are needed whenever there is a mismatch between the default baud rate of the printer port and whatever device is attached to the printer port. Each of these programs consists of just two steps:

```
363 WDVS
# WDEV
```

where # is a code for the baud rate. These codes are defined as follows:

#	BAUD RATE
6	300
7	600

10	1200
12	2400
14	4800
16	9600

PMAC.FOR This program prints a NICOS file containing multiple macros using format specifications entered from the keyboard.

!ABL
DUM. "ABL" < BSL
DUM. BLANKS ABSORBANCE FILE DFN.
TXD
FCD=-1
MUD
ABD
END

!ALN
DUM. "ALN" > SA0, M19
OMD
*** FTIR 740 ALIGNMENT MACRO ***
OMD
*
PFN=49
SPF
CSN=2
SA0
OMD
DETECTOR:
DET
OMD
APERTURE (SM, MD, OR FL):
APT
OMD
DOES THE SELECTED APERTURE DIFFER FROM THE CURRENT
OMD
PHYSICAL STATE OF THE IRIS (0=NO, 1=YS)?
CMP=0
CMP
CMP=CMP*19
DUM: CMP=0 (DUMMY) OR 19 (COLLECT 1 SCAN)
CMR
VI0=0
VI1=3
VI2=5
VI3=NDP
GFN
TMXFTR.GCP
FRN
NDP=VI3
PFN=49
SPF
END

!AOB
DUM: "AOB" > SB1
OMD

MACRO CONVERTS 1-BEAM FILE (RFN) TO "ABSORBANCE" (DFN) USING
OMD
THE MAX INTENSITY BETWEEN XSP & XEP AS THE "BACKGROUND" FOR RFN.
OMD
*
OMD
1-BEAM INPUT FILE:
RFN
OMD
ABSORBANCE OUTPUT FILE:
DFN
OMD
CM-1 RANGE OF INTEREST:
XSP
XEP
SB1
ASD
DSD
END

!ASC
OMD
"ASC" COMPRESSES A BLOCK OF SCRATCH FILES INTO AN ASCII FILE
OMD
& DIRECTS THE OUTPUT THROUGH THE PRINTER PORT TO A 2ND COMPUTER.
OMD
(1) CONNECT THE PRINTER PORTS OF THE 2 COMPUTERS
OMD
(2) SET THE NICOLET PRINTER PORT SWITCH TO "DCE"
OMD
(3) SET UP A LOG FILE ON THE HOST COMPUTER WITH "PARITY = NONE"
OMD
OF FILES (MAX=7):
QIT
OMD
1ST FILE:
DFN
PAD
OMD
1ST (FXF) & LAST (LXF) CM-1:
FXF
LXF
OMD
PRESS <CR> TO BEGIN EXECUTION
PAU
GFN
ASCSND.FCP
FRN
END

```

!ATG
DUM: "ATG" > AT1,AT2
OMD
*
OMD
*** AUTOMATIC TITLE GENERATION FOR FTIR FILES ***
OMD
*
OMD
THIS IS A DOCUMENTATION MACRO ONLY.  GENERATING TITLES REQUIRES
OMD
INSERTION OF AN IMD COMMAND AND A FRN COMMAND (FOR ATG.FCP) AT
OMD
SEPARATE LOCATIONS IN A PROGRAM THAT COLLECTS A BLOCK OF
OMD
SPECTRAL FILES.  SRT & DFN MUST BE SET TO THE INITIAL & CURRENT
OMD
FILES OF THIS BLOCK.
OMD
THE MACRO DESRIPTOR OF DFN IS USED TO STORE THE KEYBOARD INPUT
OMD
SPECIFICATIONS FOR THE TITLES OF THE SPECTRAL FILES.
OMD
*
OMD
GIVE 5 DATA FIELDS OF TITLE INFORMATION, USING ASTERISKS TO
OMD
SEPARATE THE FIELDS.  USE CONSECUTIVE ASTERISKS TO SPECIFY AN
OMD
EMPTY FIELD.  TYPING JUST [CR] IS ACCEPTABLE, AS IS OMITTING
OMD
EVERYTHING AFTER THE GENERIC FILE DESCRIPTOR.
AT1
END

```

```

!AT1
DUM: "AT1" < ATG
OMD
*
OMD
FIELD #1:  GENERIC FILE NAME
OMD
FIELD #2:  INITIAL FILE EXTENSION #
OMD
FIELD #3:  GENERIC FILE DESCRIPTOR
OMD
FIELD #4:  REFERENCE TIME FROM WHICH TO COMPUTE AN ELAPSED TIME;
OMD
           SPECIFY WITH AN 8-INTEGGER # IN ACCORDANCE WITH THE
OMD

```

FOLLOWING EXAMPLE:

OMD
*
OMD
MAY 6 11:03 BECOMES 05061103

OMD
*
OMD
FIELD #5: "D" FOR DATE
AT2
END

!AT2
DUM: "AT2" < ATG

OMD
*
OMD
EXAMPLE OF FULL TITLE SPECIFICATION:

OMD
*
OMD
HZ*1*AIR OXID*05061103*D

OMD
*
OMD
NOTE THAT THE TITLE IS RESTRICTED TO 40 CHARACTERS.
END

!BAK
DUM: "BAK" > SA0,M17,M23
OMD
*** "BAK" COLLECTS & DISPLAYS BACKGROUND SPECTRA ***

OMD
*
OMD
CHOOSE RESOLUTION CODE:
OMD

	CSN	CM-1
OMD	0	0.5
OMD	1	1
OMD	2	2
OMD	3	0.3
OMD	4	4
CSN		

SAO
OMD
DETECTOR:
DET
TEM=100
NPR
OMD
FILE #:
BFN
OMD
APERTURE (SM, MD, OR FL):
APT
DUM: AFP OPERATES ON DFN.
VIO=DFN
DFN=BFN
OMD
OF SCANS:
NSB
OMD
CM-1 LIMITS FOR SCREEN DISPLAY:
XSP
XEP
CLB
CMP=23
CMR
GFN
I.001[,IRDATA]
EXT=BFN
AFP
OMD
INTERFEROGRAM SENT TO [,IRDATA]
FPB
ASB
DSB
DFN=VIO
END

!BCR
DUM. "BCR" > BC1.
OMD
**** INTERACTIVE BASELINE CORRECTION ***
OMD
FILE #
DFN
OMD
1ST % LAST CM-1 OF THE FULL SPECTRUM:
FXF
LXF
BC1
OMD
USE X/Y ROLLS, X/Y ZOOMS, & CONTROL U TO DETERMINE CM-1 RANGES

```
OMD
TO BE CORRECTED. RECORD THE CM-1 FOR LATER USE. THEN HIT [CR].
PAU
OMD
# OF CM-1 RANGES TO BE CORRECTED:
QIT
FOR AAA=1 TIL QIT
OMD
INITIAL & FINAL CM-1 OF A SEGMENT TO BE CORRECTED:
XSP
XEP
OMD
USE X-ROLL, Y-ROLL, & Y-ZOOM ON BASELINE. SAVE WITH CONTROL Z
BLC
VFO=XSP
XSP=XEP
XEP=VFO
NXT AAA
BC1
END
```

```
!BC1
DUM. "BC1" < BCR
XSP=FXF
XEP=LXF
ASD
DSD
TEM=100
NPR
END
```

```
!BLX
DUM. "BLX"
OMD
*** SPECTRUM BLANKING MACRO ***
FXF=XSP
LXF=XEP
TEM=100
NPR
OMD
FILE TO UNDERGO PARTIAL BLANKING:
DFN
PAD
OMD
# OF CM-1 RANGES TO BE BLANKED:
QIT
GFN
BLANKX.FCP
FOR III=1 TIL QIT
```

OMD
ENTER A PAIR OF BLANKING RANGE LIMITS (CM-1):
XSP
XEP
FRN
NXT III
XSP=FXF
XEP=LXF
ASD
DSD
END

!BLZ
OMD
"BLZ" BLANKS SELECTED PORTIONS OF AN ABSORBANCE FILE BY SETTING
OMD
ABS=0 & USES BASELINE ADJUSTMENT TO AVOID SUDDEN OFFSETS IN THE
OMD
Y-AXIS AT CM-1 RANGE LIMITS. SEE DOCUMENTATION IN "BLANKZ.FOR"
OMD
FOR SETUP OF THE "BLANK.DAT" NICOS INPUT DATA FILE.
OMD
INPUT (SFN) & OUTPUT (DFN) FILES:
SFN
DFN
OMD
DATA SET # OF "BLANK.DAT" FILE:
VI2
MSD
PAD
TXD
GFN
BLANKZ.FCP
FRN
ABD
END

!BSL
DUM. "BSL" > BS0-BS2,ABL,M20-21
OMD
AUTOMATED BASELINE-CORRECTION PROGRAM "BSL".
BS0
DUM. 'MRD' ENSURES VALID FSB'S:
MRD
MNT=DFN
DFN=OFN
MRD
ABL. BLANK FILE OFN
DFN=MNT
GFN

BASLIN.FCP
FRN
BS1
BS2
END

!BS0
DUM. "BS0" < BSL
OMD
USE EXISTING "BASLIN.DAT" (VI1=0) OR CREATE FILE (VI1=1)?
VI1
CMP=VI1*21
CMR
OMD
ABSORBANCE INPUT FILE:
RFN=5
PRN RFN
OMD
OUTP FOR ABSORBANCE (DFN) & BASELINE PT DISPLAY (OFN):
DFN=c
PRN DFN
OFN=0
PRN OFN
PAR
OMD
1ST & LAST CM-1:
FXF=XSP+1
LXF=XEP-1
PRN FXF
PRN LXF
OMD
RESET INPUT (0 OR 1)?
CMP=0
CMP
CMP=20*CMP
CMR
END

!BS1
DUM. "BS1" < BSL
DUM. COMPARES SUCCESSIVE FRAGMENTS OF DFN & RFN USING CONSTANT
DUM. ABS/INCH (CF. "DSC"), ALONG WITH FILE OFN WHICH CONTAINS
DUM. BASELINE REF PTS GENERATED BY "BASLIN.FOR". THE DISPLAY
DUM. BEGINS AT XSP.
VI1=SFN
VI2=XSP
VI3=XEP
OMD

```
SFN=OFN
OMD
CM-1 RANGE PER SCREEN DISPLAY (- SIGN FOR DECREASING CM-1):
VF1=-400
VF1
XEP=XSP+VF1
OMD
# OF DISPLAY RANGES:
QIT=9
QIT
TEM=9
FOR III=1 TIL QIT
BS2
NXT III
SFN=VI1
XSP=XSP-VF1
XEP=XEP-VF1
END
```

```
!BS2
DUM. "BS2" <BSL
ASR
DSR
VFO=YEP-YSP
TEM=9
NPR
ASD
YEP=YSP+VFO
DSD
OMD
HIT 'RETURN' TO CONTINUE
PAU
NPR
ASS
DSS
OMD
HIT 'RETURN' TO CONTINUE
PAU
TEM=5
NPR
XSP=XSP+VF1
XEP=XEP+VF1
END
```

```
!DCP
OMD
"DCP" COMPARES SEVERAL SPECTRA WHICH HAVE BEEN SCALED SO THAT A
OMD
SPECIFIED REF PEAK HAS THE SAME SCREEN HEIGHT IN ALL SPECTRA.
```

OMD
*
OMD
OF FILES TO BE COMPARED:
VIO
OMD
CM-1 LIMITS FOR THE AUTOSCALING REFERENCE PEAK:
VFO
VF1
OMD
OF CM-1 RANGES TO BE DISPLAYED:
QIT
FOR AAA=1 TIL QIT
OMD
ENTER A PAIR OF CM-1 RANGE LIMITS:
XSP
XEP
VF2=XSP
VF3=XEP
FOR III=1 TIL VIO
OMD
FILE #:
DFN
CSN=DFN
XSP=VFO
XEP=VF1
ASD
XSP=VF2
XEP=VF3
YSP=0
DSD
OMD
SHIFT THE BASELINE IF NECESSARY; THEN HIT "RETURN."
PAU
TEM=9
NPR
DFN=CSN
NXT III
TEM=5
NPR
NXT AAA
END

!DER
OMD
"DER" DERESOLVES AN ABSORBANCE FILE TO 16 CM-1 RESOLUTION
OMD
*
PFN=49
SPF
OMD

OF FILE TO BE DERESOLVED:

DFN
ASD
DSD
TEM=9
NPR
NTP=2048
PTS=5
GFN
RESOLU.FTN
OFN=DFN
FRN
DSD
RPF
END

!DSC

DUM. "DSC" > DS1
DUM. COMPARES SUCCESSIVE FRAGMENTS OF 2 FILES USING CONSTANT
DUM. ABS/INCH.
DUM. YSP IS OBTAINED SEPARATELY FOR RFN & DFN BY AUTOSCALING.
DUM. YEP IS OBTAINED FOR RFN BY AUTOSCALING; & FOR DFN BY
DUM. ADDING THE RANGE OF RFN TO THE YSP OF DFN.

OMD

AUTO-SCALED (RFN) & COMPARISON (DFN) FILES:

RFN
DFN
OMD

STARTING CM-1 TO BE DISPLAYED:

XSP

OMD

CM-1 RANGE PER DISPLAY (- FOR DECREASING CM-1):

VF1

XEP=XSP+VF1

OMD

OF DISPLAY RANGES:

QIT

TEM=9

FOR III=1 TIL QIT

DS1

NXT III

XSP=XSP-VF1

XEP=XEP-VF1

END

!DS1

DUM. "DS1"<DSC

ASR

DSR

VFO=YEP-YSP
TEM=9
NPR
ASD
YEP=YSP+VFO
DSD
OMD
HIT 'RETURN' TO CONTINUE
PAU
TEM=5
NPR
XSP=XSP+VF1
XEP=XEP+VF1
END

!DUM
END

!EMP
DUM. "EMP" > M99
OMD
"EMP" COPIES A SPECTRAL FILE INTO ALL VACANT FILES IN A BLOCK
OMD
OF SCRATCH FILES. THIS PREVENTS AN 'INVALID FSB' MESSAGE FROM
OMD
OCCURRING DURING EXECUTION OF MACRO "TTL".
OMD
*
OMD
FILE TO BE COPIED:
OFN
OMD
1ST (SRT) & LAST (QIT) FILE #'S OF BLOCK:
SRT
QIT
DFN=SRT-1
FOR III=SRT TIL QIT
DFN=DFN+1
GFN
EMPFIL.FCP
FRN
CMR
NXT III
END

!FSP
OMD

*** "FSP" PRINTS A SECTOR OF FTIR SCRATCH FILE DATA ***
DUM. TO PRINT THE FILE STATUS BLOCK, REQUEST SECTOR FSZ/512
OMD
ENTER FILE #:
DFN
OMD
ENTER SECTOR # (HERE, THE 1ST SECTOR IS CALLED #1, NOT #0):
PTS
GFN
FSPRN.FCP
FRN
END

!FSW
DUM. "FSW" WRITES A WORD INTO THE STATUS BLOCK OF AN FTIR FILE
OMD
ENTER FILE #:
DFN
PAD
OMD
ENTER WORD # (HERE, THE WORD IS CALLED #1, NOT #0):
VIO
OMD
ENTER NEW VALUE OF WORD:
V11
GFN
FSWR.FCP
FRN
END

!ITG
OMD
"ITG" INTEGRATES SELECTED CM-1 RANGES OF AN ABSORBANCE FILE.
OMD
SEE DIRECTIONS IN "INTGRL.FOR" FOR SETUP OF THE "INTSEG.DAT"
OMD
NICOS INPUT DATA FILE.
OMD
*
BAS=YS
GFN
INTGRL.FCP
OMD
DATA SET # OF "INTSEG.DAT":
VI2
OMD
NUMBER OF FILES:
VIO
OMD

INITIAL (VI3) & FINAL (QIT) CM-1 RANGE INDICES:

VI3=1

VI3

QIT

QIT=QIT+1

OMD

REFERENCE CM-1 RANGE INDEX:

VI1

OFN=VI1

FOR FFF=1 TIL VIO

OMD

INPUT FILE #:

DFN

TEM=4

NPR

TIQ

VI1=VI3-1

FRN

SMD

VFO=FCD

FOR III=VI3 TIL QIT

VI1=VI1+1

FRN

SMD

NXT III

TEM=4

NPR

NXT FFF

END

!LBB

OMD

"LBB" AUTO-TITLES A BLOCK OF SCRATCH FILES & TRANSMITS IT TO A

OMD

USER-SPECIFIED DISK LOCATION.

OMD

*

LB1

OMD

CM-1 RANGE TO BE SAVED (FXF, LXF):

FXF

LXF

DUM. SRT USED IN AUTIT.FCP

SRT=DFN

DFN=SRT-1

OMD

GENERIC FILE TITLE (SEE "ATG" MACRO FOR SPECIAL TITLE OPTIONS):

IMD

DFN=SRT

RTN=0

FOR AAA=1 TIL QIT

```

PRN DFN
GFN
AUTIT.FCP
FRN
TIQ
DFN=DFN+1
NXT AAA
DFN=SRT
OMD
OUTPUT FILES: NAME, DUMMY EXT, DEVICE, & DIR
OMD
E.G. ABS.000-D1[DATA]:
OMD
*
IFN
OMD
1ST EXTENSION # FOR OUTPUT:
EXT
FOR AAA=1 TIL QIT
PRN DFN
TIQ
ASP
EXT=EXT+1
DFN=DFN+1
NXT AAA
DFN=DFN-1
END

```

```

!LBI
DUM. "LBI" > LB1, LB2
OMD
*
OMD
"LBI" MOVES A BLOCK OF "I.EXT[,IRDATA]" INTERFEROGRAM FILES
OMD
CREATED BY MACROS "BAK", "STI", OR "SAM" TO A USER-SPECIFIED
OMD
LOCATION AFTER INITIALLY TRANSFERRING THEM TO A BLOCK OF FTIR
OMD
FILES.
OMD
*
GFN
I.000[,IRDATA
OMD
1ST EXTENSION # OF "I.EXT" FILES:
EXT
LB1
OFN=DFN
FOR III=1 TIL QIT
PRN DFN

```

AFG
DFN=DFN+1
EXT=EXT+1
NXT III
DFN=OFN
TIQ
PAD
FXI=0
LXI=NDP
LB2
END

!LBM
DUM. "LBM" > LB1, LB2
OMD
*
OMD
"LBM" COPIES A BLOCK OF SCRATCH FILES TO A USER-SPECIFIED DISK
OMD
LOCATION
OMD
*
LB1
OMD
CM-1 RANGE TO BE SAVED (FXF, LXF):
FXF
LXF
OMD
*
TIQ
LB2
END

!LBW
DUM. "LBW" > LB1, LB2
OMD
*
OMD
"LBW" COPIES A BLOCK OF WAVELENGTH FILES TO A USER-SPECIFIED
OMD
DISK LOCATION.
OMD
*
LB1
OMD
WAVELENGTH RANGE TO BE SAVED (FXL, LXL):
FXL
LXL
OMD

*
TIQ
LB2
END

!LB1
DUM. "LB1" < LBI & LBM
OMD
OF FILES:
QIT
OMD
1ST SEQUENTIAL SCRATCH FILE:
DFN
END

!LB2
DUM. "LB2" < LBI & LBM
OMD
*
OMD
OUTPUT FILES: NAME, DUMMY EXT, DEVICE, & DIR
OMD
E.G. ABS.000-D1[DATA]:
OMD
*
IFN
OMD
1ST EXTENSION # FOR OUTPUT:
EXT
FOR III=1 TIL QIT
PRN DFN
TIQ
ASP
DFN=DFN+1
EXT=EXT+1
NXT III
END

!LTR
OMD
"LTR" TRANSMITS TEXT FROM THE KEYBOARD TO THE PLOTTER. THE SIZE
OMD
& ORIENTATION OF THE LETTERS ARE USER SPECIFIED.
OMD
*
GFN
LETTRS.FCP

FRN
END

!MMM
END

!MMP
END

!MOO
END

!M12
DUM. "M12" < SAM,STI
OMD
MINUTES FROM FINISH OF DATA CRUNCH TO START OF NEXT SCAN:
VI2
VIO=VI2*30
END

!M13
DUM. "M13" < SAM,STI
OMD
* * * * * DELAY IN PROGRESS * * * * *
DCL=212234
DCX
TEM=11
NPR
DCL=208138
DCX
END

!M14
DUM. "M14" < SAM,STI
END

!M15
DUM. "M15" < SAM,STI
OMD

HIT "RETURN" KEY WHEN READY TO CONTINUE.
PAU
END

!M17
DUM. "M17" < SA0 < BAK,SAM,STI,ALN
PFN=CSN
GFN
BENPAR.740
APG
RPF
END

!M19
DUM. "M19" < ALN
NSS=1
CLS
END

!M20
DUM. "M20" < BSL
OMD
ABSORBANCE INPUT FILE:
RFN
OMD
OUTPUT FILES FOR ABSORBANCE (DFN) & BASELINE PT DISPLAY (OFN):
DFN
OFN
OMD
1ST & LAST CM-1 OF CORRECTION RANGE:
FXF
LXF
END

!M21
DUM. "M21" < BSL
OMD
OF DATA PTS PER BASELINE PT:
VIO=128
VIO
OMD
CM-1 RANGE LIMITS TO SKIP: (DSK,ITR), (RTN,PTS), (NTR,PEK).
OMD
SET UNUSED LIMITS TO ZERO:
DSK=0

DSK
ITR=0
ITR
RTN=0
RTN
PTS=0
PTS
NTR=0
NTR
PEK=0
PEK
END

!M23
DUM. "M23" < STI
OMD
FILE TITLE:
TID
END

!M30
DUM. "M30" < PLC
OMD
OF INCHES OF Y-AXIS DEAD-SPACE BETWEEN PLOTS:
VF4=0.2
VF4
CMP=VI2/2
VF4=CMP*.01*SRM*VI2+VF4
PRN VF4
END

!M31
DUM. "M31" < PLC
OMD
Y-AXIS BASELINE OFFSET BETWEEN PLOTS (INCHES):
VF4
END

!M40
DUM PLC & PLI > "M40" (SET ANGLES FOR A NONROTATED PLOT)
DUM. ANGLES OF X (SRT) & Y (RTN) AXES:
SRT=0
RTN=270
DUM. ANGLES OF X (FCD) & Y (RTO) TIC LABELS:

FCD=0
RTO=0
DUM. ANGLES OF X (XSL) & Y (YSL) AXIS NAMES:
XSL=0
YSL=90
END

!M41
DUM. PLC & PLI > "M41" (SET ANGLES FOR A ROTATED PLOT)
DUM.
DUM. TEP LYT FXL SRT RTN FCD RTO XSL YSL
DUM. NORM ROT: - - + 90 0 90 180 90 180
DUM. ANTI ROT: + + - 270 180 270 270 270 0
DUM.
DUM. NORM ROT CAUSES "LINE" SUBROUTINE TO FAIL.
DUM.
DUM. ANGLES OF X (SRT) & Y (RTN) AXES:
SRT=270
RTN=180
DUM. ANGLES OF X (FCD) & Y (RTO) TIC LABELS:
FCD=270
RTO=270
DUM. ANGLES OF X (XSL) & Y (YSL) AXIS NAMES:
XSL=270
YSL=0
DUM. X AXIS IS IN - DIRECTION FOR A ROTATED PLOT:
~XL=-FXL
END

!M42
DUM. "M42" < PLC, PLI
ASD
DSD
OMD
1ST & LAST Y AXIS LIMITS:
YSP
YEP
OMD
1ST TIC LABEL ON Y AXIS:
FYI
OMD
TIC LABEL SPACING FOR Y AXIS:
CYA
END

!M43
DUM. "M43" < PLC, PLI

RFN=DFN
ASR
END

!M44
DUM. "M44" < PLC, PLI
RFN=SMN
ASR
END

!M45
DUM. "M45" < PLC, PLI
OMD
AUTOSCALE LIMITS FOR X AXIS:
XSP
XEP
WTY=-15
END

!M46
DUM. PLC & PLI > "M46" (SCALE PARAMETERS) > M45
OMD
1ST & LAST X VALUES TO BE PLOTTED:
FXF
LXF
XSP=FXF
XEP=LXF
OMD
1ST TIC LABEL ON X AXIS:
LXI
OMD
SPACING OF TIC LABELS ON X AXIS:
CXL
OMD
LENGTHS (INCHES) OF X (FXL) & Y (LYT) AXES:
FXL
LYT
OMD
AUTOSCALING: 0=NONE, 1=ALL, >1 = AUTOSCALE ONCE USING FILE SMN
SMN=1
SMN
DUM: CALL M45 (AUTOSCALE LIMITS) OR MMM (NOTHING)
CMP=SMN/10000
CMP=46*CMP-1
CMR
DUM. MARGIN (1/100 INCHES) BETWEEN AUTOSCALED SPECTRUM BASELINE
DUM. & X AXIS:

MNT=10
END

!M47
DUM. PLC & PLI > "M47" (MISC INPUT PARAMETERS > (M30-31)
OMD
OF PLOTS:
QIT
OMD
PLOT TITLES (0=NO, 1=YES)? NOTE: TITLE OFFSET = PLOT OFFSET
VI2
OMD
INITIAL PEN # FOR SPECTRAL DATA:
PEK=1
PEK
OMD
SPECTRUM PEN # INCREMENT FOR SUCCESSIVE PLOTS:
NSS=0
NSS
OMD
OVERLAP Y-AXES (0=NO, 1=YES)?
OFN=OFN/10000
OFN
OFN=OFN/10000
CMP=OFN+30
CMR
END

!M48
DUM. PLC & PLI > "M48" > (M40,41)
OMD
ROTATE PLOT (0=NO, 1=YS)?
RTR=RTR/10000
RTR
OMD
ORIGIN (INCHES) OF X (LXL) & Y (LYA) AXES VS. ZPN:
OMD
(NOTE: FOR A ROTATED PLOT, LXL ~ 1 + X-AXIS LENGTH)
LXL=FXL*RTR+1
LXL
LYA
CMP=RTR/10000+40
CMR
OMD
RATIO OF MINOR TO MAJOR TICS:
RTP
GFN
PLPARW.FCP
FRN

END

!M49
DUM. PLC & PLI > "M49"
GFN
PLPARR.FCP
FRN
END

!M50
DUM. SA5 < "M50" < SAM,STI
BFN
PAB
OMD
DETECTOR:
DET
OMD
APERTURE (SM, MD, OR FL)
APT
OMD
ENTER 12 FOR AUTO-TIMED SPECTRA; 14 FOR MANUAL:
VI1
CMP=VI1
CMR
OMD
OF SCANS PER SPECTRUM:
NSD
OMD
OF SPECTRA TO BE COLLECTED
QIT
SA5
END

!M51
DUM. "M51" < SAM
BFN=3
PRN BFN
PAB
DET=1
PRN DET
APT=MD
PRN APT
VI1=12
OMD
PROGRAM PAUSE (MIN) BETWEEN AUTO-SCANS:
VI2=60
PRN VI2

VI0=VI2*30
OMD
SCANS PER SPECTRUM:
NSD=500
PRN NSD
OMD
OF SPECTRA TO BE COLLECTED:
QIT=20
PRN QIT
OMD
1ST & LAST CM-1 FOR 1ST DISPLAY:
VF0=4000
VF1=600
PRN VF0
PRN VF1
OMD
1ST & LAST CM-1 FOR 2ND DISPLAY:
VF2=1200
VF3=1650
PRN VF2
PRN VF3
END

!M52
DUM. "M52" < SAM,STI
BFN=3
PRN BFN
PAB
DET=1
PRN DET
APT=MD
PRN APT
VI1=14
OMD
SCANS PER SPECTRUM:
NSD=100
PRN NSD
OMD
OF SPECTRA TO BE COLLECTED:
QIT=40
PRN QIT
OMD
1ST & LAST CM-1 FOR 1ST DISPLAY:
VF0=4000
VF1=600
PRN VF0
PRN VF1
OMD
1ST & LAST CM-1 FOR 2ND DISPLAY:
VF2=1200
VF3=1700

PRN VF2
PRN VF3
END

!M99
DUM. "M99" < EMP
MOD
END

!NSF
OMD
*** "NSF" COPIES A BLOCK OF FILES FROM NICOS TO FTIR ***
OMD
*
OMD
GENERIC FILE NAME & LOCATION:
OMD
E.G. FILNAM OR FILNAM[DIR] OR FILNAM.000-D1[DIR]
IFN
OMD
1ST EXTENSION #:
EXT
OMD
1ST SCRATCH FILE #:
DFN
OMD
OF FILES TO COPY:
QIT
FOR III=1 TIL QIT
PRN DFN
AFG
DFN=DFN+1
EXT=EXT+1
NXT III
DFN=DFN-1
END

!NXP
OMD
NEXT PAGE ADVANCE FOR ZETA-8 PLOTTER
SF2=RO
YPN=-8.5
XPN=0
PEN
YPN=0
ZPN
SF2=BT

END

```
!PCP
DUM: "PCP" > PC1
PC1
OMD
1) SET PEN AT PERFORATION; 2) [CR]
PAU
XPN=1
YPN=1
PEN
FOR AAA=1 TIL QIT
DFN=VI1-1
ZPN
AXS=XO
OMD
PLOTING RANGE:
FXF
LXF
OMD
CM-1 PER INCH (NEG IF FXF>LXF ):
CXF
FXL=LXF-FXF
FXL=FXL/CXF
FOR III=1 TIL VIO
DFN=DFN+1
XSP=VF0
XEP=VF1
ASD
YSP=0.0
YEP=YEP/VF3
CYA=YEP/VF2
XSP=FXF
XEP=LXF
DSD
OMD
1) USE CURSOR TO RAISE BASELINE FULLY INTO VIEW; 2) [CR]
PAU
TEM=9
NPR
FYA=YSP
LYA=YEP
TEM=103
NPR
PLD
PCL=CSN+PCL
TEM=100
NPR
XPN=0
YPN=VF4
PEN
```

AXS=NO
NXT III
PCL=0
PL6
TEM=5
NPR
NXT AAA
END

!PC1
DUM: "PC1" < PCP
OMD
"PCP" COMPARES A BLOCK OF SPECTRA BY EQUALIZING THE ABSORBANCES
OMD
OF A SELECTED BAND. MULTIPLE CM-1 RANGES CAN BE SPECIFIED.
OMD
*
OMD
AUTOSCALE LIMITS FOR REFERENCE BAND (XSP, XEP):
XSP
XEP
VF0=XSP
VF1=XEP
OMD
RATIO OF AUTOSCALED PEAK ABSORBANCE TO YEP:
VF3
OMD
TITLES (YS OR NO)?
TIT
OMD
AUTO PEN CHANGE (0=NO, 1=YS)?
CSN=0
CSN
OMD
OF CM-1 RANGES TO BE PLOTTED:
QIT
OMD
TOTAL # OF SEQUENTIAL SCRATCH FILES:
VIO
OMD
Y-AXIS OFFSET BETWEEN PLOTS:
VF4=1
VF4
VF2=10-VIO
OMD
1ST FILE #:
VII
SF2=NO
RTR=0
LYT=0
END


```
!PFB
DUM. "PFB" < R#W & M17(ALN,BAK,STI,SAM)
DUM. "PFB" READS PARAMETER FILE PFN OF "BENPAR.740"
GFN
BENPAR.740
APG
RPF
PRN NDP
PRN NTP
PRN FSZ
PRN MIR
PRN LPS
PRN HPS
PRN VEL
PRN XSP
PRN XEP
PRN SFN
PRN BFN
PRN DFN
PRN RFN
PRN NSS
PRN NSB
PRN NSD
PRN APT
PRN DET
END
```

```
!PLC
DUM: "PLC" > PL1-PL4,PL7,M(30-31,40-49)
OMD
"PLC" COMPARES SEVERAL PLOTS ON 1 X-AXIS *****
PFN=49
SPF
TEM=100
NPR
PL1: DEFAULT TITLE    PARAMETERS
PL2:      "  AXES      "
PL3:      "  TIC       "
PL4:      "  TIC LABEL  "
OMD
NICOS (0) OR MANUAL (1) INPUT ?
BFN=BFN/1000
BFN
BFN=BFN/1000
CMP=51*BFN-5
DUM: SET SCALE PARAMS (M46 OR MMM)
CMR
CMP=51*BFN-4
DUM: MISC INPUT (M47 OR MMM)
```

```
CMR
CMF=51*BFN-3
DUM: MISC INPUT (M48 OR MMM)
CMR
CMP=51*BFN+49
DUM: READ "PLT.DAT" PARAMS; (M49 OR MMP)
CMR
PL7: PLOTTING LOOP.
PFN=49
RPF
END
```

```
!PLI
DUM: "PLI" > PL1-PL6,M(40-46,48,49)
OMD
*** "PLI" PLOTS N FILES ON N X-AXES ***
OMD
WAIT FOR PROMPTS
PFN=49
SPF
TEM=100
NPR
PL1: DEFAULT TITLE   PARAMETERS
PL2:      "    AXES      "
PL3:      "    TIC      "
PL4:      "    TIC LABEL  "
OMD
NICOS (0) OR MANUAL (1) INPUT ?
BFN=BFN/1000
BFN
BFN=BFN/1000
CMP=51*BFN-5
DUM: SET SCALE PARAMS (M46 OR MMM)
CMR
CMP=51*BFN-3
DUM: MISC INPUT (M48 OR MMM)
CMR
CMP=51*BFN+49
DUM: READ "PLT.DAT" PARAMS; (M49 OR MMP)
CMR
PL5: PLOTTING LOOP.
PFN=49
RPF
END
```

```
!PL1
DUM PLC & PLI > "PL1" (DEFAULT TITLE PARAMETERS)
DUM. PLOT TITLE (0=NO, 1=YES, 2=USE MACRO DESCRIPTOR AS TITLE)?
VI2=1
```

DUM. INCHES FROM ORIGIN OF X AXIS TO 1ST LETTER OF TITLE
DUM. MEASURED ALONG X AXIS (NEGATIVE # GIVES AUTO-CENTERING):
FXI=0
DUM. INCHES FROM UPPER LIMIT OF Y AXIS TO 1ST LETTER OF TITLE
DUM. MEASURED ALONG Y AXIS:
LYE=0.1
DUM. SIZE OF TITLE LETTERS IN 1/100 INCHES:
SRM=15
DUM. PEN # FOR TITLE:
VI3=1
END

!PL2
DUM. PLC & PLI > "PL2" (DEFAULT AXES PARAMETERS)
DUM. PLOT X AXIS (0=NO, 1=YES)?
VIO=1
DUM. PLOT Y AXIS (0=NO, 1=YES)?
VI1=1
DUM. SIZE OF LETTERS OF AXES NAMES IN 1/100 INCHES (A NEGATIVE
DUM. VALUE KILLS THE AXES NAMES):
SRN=15
DUM. PEN # FOR AXES:
PTS=1
END

!PL3
DUM. PLC & PLI > "PL3" (DEFAULT TIC PARAMETERS)
DUM. MAJOR TIC SIZE IN 1/100 INCHES:
DSK=10
DUM. SIZE RATIO OF MAJOR TO MINOR TICS:
FCB=2.0
DUM. # RATIO OF MINOR TO MAJOR TICS FOR X (RTP) & Y (RTQ) AXES:
RTP=1
RTQ=0
DUM. MARGIN (1/100 INCHES) BETWEEN TIC LABELS & TICS:
SIZ=10
DUM. MARGIN (1/100 INCHES) BETWEEN TIC LABELS & AXIS NAME:
NSR=20
END

!PL4
DUM. PLC & PLI > "PL4" (DEFAULT TIC LABEL PARAMETERS)
DUM. DIGIT SIZE (1/100 INCHES) FOR X (ITR) & Y (WTY) TIC
DUM. LABELS; - SIGN KILLS TICS & TIC LABELS:
ITR=15
WTY=15
DUM. IF TIC LABEL & AXIS ARE PARALLEL, GIVE # OF CHARACTERS OF

DUM. TIC LABEL TO LIE TO THE RIGHT OF THE X (VF2) OR Y (VF3)
DUM. TIC MARK.
VF2=1.6
VF3=1.6
DUM. # OF DIGITS TO RIGHT OF DECIMAL POINT FOR X (NTR) & Y (PFN)
DUM. AXES; PFN > 10 GIVES AUTO-CALCULATED VALUE FOR Y AXIS:
NTR=0
PFN=11
END

!PL5
DUM. PLI > "PL5" > M42-44
DUM: SET Y-AXIS MACRO (M42, M43, OR M44)
CMP=SMN+9999
CMP=CMP/10000-1
CMP=SMN/10000+CMF+42
OMD
TITLES (0=NO, 1=YES)?
VI2
DUM. TURN OFF OVERLAP PARAMETERS (OFN & VF4):
OFN=0
VF4=-11
OMD
OF PLOTS:
QIT
ZPN
PAU SET PEN AT PERFORATION <CR>
FOR III=1 TIL QIT
PEK=1
OMD
FILE #:
DFN
PAD
ZPN
ASD
DSD
CMR
GFN
PLOTFT.FCP
FRN
PL6
NXT III
END

!PL6
DUM. "PL6" < PLI & PCP *** SET NEW PAGE ON PLOTTER ***
DUM. FXL & LYT ARE THE X AND Y AXIS LENGTHS, RESPECTIVELY.
DUM. RTR DENOTES NONROTATED (0) OR ROTATED (1) PLOT.
DUM. INITIALLY, XPN & YPN ARE USED AS TEMPORARY JUNK VARIABLES.

```
XPN=FXL+.1
YPN=1-RTR
YPN=YPN*XPN/8.5
XPN=LYT+.1
XPN=RTR*XPN/8.5
DUM. PEK IS THE TRUNCATED INTEGER SUM OF REAL #'S
PEK=XPN+YPN
XPN=8.5*PEK
YPN=0
PEN
END
```

```
!PL7
DUM. PLC > "PL7" > (M42-44)
DUM: SET Y-AXIS MACRO (M42, M43, OR M44)
CMP=SMN+9999
CMP=CMP/10000-1
CMP=SMN/10000+CMP+42
ZPN
SFN=SMN/10000
DTD=0
DUM.
DUM. *** BEGIN PLOTTING LOOP ***
DUM.
FOR AAA=1 TIL QIT
OMD
FILE # FOR PLOT:
DFN
PAD
PLO=300
VI3=PEK
DUM. SET Y SCALE:
CMR
GFN
PLOTFT.FCP
FRN
DUM. AXES(VI0,VI1) & Y-INPUT(CMP) ON/OFF:
VI0=0
VI1=1-OFN
CMP=SFN*CMP
DTD=DTD+1
PEK=PEK+NSS
NXT AAA
END
```

```
!PMY
OMD
*** "PMY" PRINTS Y-AXIS LIMITS FOR A BLOCK OF FTIR FILES ***
OMD
```

```

1ST FILE #:
SRT
OMD
LAST FILE #:
QIT
DFN=SRT
DUM: USE SUB-BLOCKS OF SIZE RTN FOR SCREEN DISPLAY:
RTN=8
RTO=1
VI1=QIT-SRT+1
VI2=VI1
VI3=VI2/RTN
FOR III=1 TIL VI3
DUM: CALC UPPER INDEX (RTQ) FOR SUB-BLOCK:
RTO=RTO+1
RTP=RTO/VI3
RTP=2-RTP
RTQ=1-RTP
RTP=RTN*RTP
RTQ=VI2*RTQ+RTP
FOR JJJ=1 TIL RTQ
OMD
*****
ASD
DSD
PRN DFN
PRN YSP
PRN YEP
DFN=DFN+1
NXT JJJ
VI2=VI2-RTN
OMD
*****
PAU/ TYPE [CR] TO CONTINUE
NXT III
END

```

```

!PPP
DUM. "PPP" > PP1
OMD
"PPP" PLOTS & PEAK-PICKS AN FTIR SCRATCH FILE
PP1
OMD
FILE #:
DFN
ASD
DSD
OMD
1ST & LAST Y-AXIS VALUES:
FYA
LYA

```

OMD
ABSORBANCE UNITS PER INCH:
CYA
OMD
PPK THRESHOLD VALUE:
THR
OMD
AXIS OPTION (YS, NO, XO):
AXS
OMD
1) SET PEN AT NEAR (SF2=NO) OR FAR (SF2=RO) PAPER PERFORATION
OMD
2) [CR]
PAU
ZPN
PEN
TEM=103
NPR
PLP
XPN=0-XPN
YPN=0-YPN
PEN
TEM=100
NPR
YPN=1
XPN=0-XPN
END

!PP1
DUM. "PP1" < PPP
TEM=100
NPR
PLO=0
FIT=YS
TIT=NO
PUP=YS
OMD
1ST & LAST CM-1 (NOTE: AXIS LENGTH MUST BE AN INTEGER):
FXF
LXF
XSP=FXF
XEP=LXF
OMD
CM-1 PER INCH (NEGATIVE FOR FXF>LXF):
CXF
OMD
SMOOTHING OPTION (YS, NO):
SMO
OMD
RESET 'PTS' IF SMOOTHING IS DESIRED (ALLOWED VALUES = 3-25)
PTS=3

PTS
OMD
PLOT ROTATION? (RO, NO):
SF2
OMD
SHIFTS (INCHES) IN THE X-AXIS (XPN) & Y-AXIS (YPN)
OMD
ORIGINS VS THE ZPN POINT:
XPN
YPN
END

! PR1
GFN
PARAM1.GCP
FRN
END

! PR2
GFN
PARAM2.GCP
FRN
END

! PR3
GFN
PARAM3.GCP
FRN
END

! PR4
GFN
PARAM4.GCP
FRN
END

! PR5
GFN
PARAM5.GCP
FRN
END


```
!RES
OMD
*** "RES" CHANGES THE RESOLUTION OF AN ABSORBANCE FILE ***
OMD
INPUT FILE #:
OFN
OMD
OUTPUT FILE #:
DFN
OMD
RESOLUTION OF OUTPUT FILE (CM-1):
VF1
NTP=32768/VF1
PRN NTP
GFN
RESOLU.FTN
FRN
END
```

```
!ROW
DUM. "ROW" > PFB
OMD
SETTING RESOLUTION TO 0.5 CM-1
PFN=0
PFB
END
```

```
!R1W
DUM. "R1W" > PFB
OMD
SETTING RESOLUTION TO 1 CM-1
PFN=1
PFB
END
```

```
!R2W
DUM. "R2W" > PFB
OMD
SETTING RESOLUTION TO 2 CM-1
PFN=2
PFB
END
```

```
!R3W
DUM. "R3W" > PFB
OMD
SETTING RESOLUTION TO 0.3 CM-1
PFN=3
PFB
END
```

```
!R4W
DUM. "R4W" > PFB
OMD
SETTING RESOLUTION TO 4 CM-1
PFN=4
PFB
END
```

```
!SAM
DUM: "SAM" > SA(0,2-5,9),M(12-15,17,50-52)
OMD
*** "SAM" COLLECTS, DISPLAYS, & AUTO-TITLES SAMPLE SPECTRA ***
OMD
*
OMD
INPUT OPTION: 50 = KEYBOARD INPUT
OMD
                    51 = AUTOSCAN WITH DEFAULT PARAMETERS FROM M51
OMD
                    52 = MANUAL SCAN WITH DEFAULT PARAMETERS FROM M52
CMP=51
CMP
TEM=100
NPR
CMR
OMD
1ST SPECTRAL FILE # (FILE DFN-1 MUST HAVE VALID FSB):
DFN
DUM. SRT USED IN AUTIT.FCP
SRT=DFN
DFN=SRT-1
OMD
GENERIC FILE TITLE (SEE "ATG" MACRO FOR SPECIAL TITLE OPTIONS):
IMD
DFN=SRT
SFN=2
RTN=0
OMD
*
OMD
HIT "RETURN" TO START
```

PAU
SA2
END

!SA0
DUM. M17 < "SA0" < BAK & SAM & STI & ALN
OMD

DUM: THE "SA0" MACRO SETS PFN=CSN (VIA "M17") IF PFN & CSN ARE
DUM: NOT EQUAL. OTHERWISE, "SA0" DOES NOTHING.
CMP=PFN-CSN
CMP=CMP/100
MNT=CSN-PFN
CMP=MNT/100+CMP
CMP=18*CMP-CMP
CMR
END

!SA2
DUM. "SA2" < SAM,STI
FOR AAA=1 TIL QIT
SA3: COLLECT INTERFEROGRAM INTO DFN & COPY IT TO I.EXT[,IRDATA].
DUM: THEN TRANSFORM & DISPLAY DATA.
SA4: DISPLAY 1ST CM-1 RANGE OF CURRENT & PREVIOUS RFN.
OMD
FILE & STARTING TIME FOR MOST RECENTLY PROCESSED SPECTRUM:
PRN DFN
STD
CMP=23*RTN
CMR
CMP=VI1+1
DUM: AUTO-DELAY (M13) OR MANUAL PAUSE (M15).
CMR
DUM: RESET CM-1 LIMITS & CLEAR SCREEN.
XSP=VF2
XEP=VF3
TEM=5
NPR
ASD
SA4: DISPLAY 2ND CM-1 RANGE OF CURRENT & PREVIOUS SPECTRUM
CMR
DFN=DFN+1
NXT AAA
DFN=DFN-1
END

!SA3

DUM. "SA3" < SAM,STI
XSP=VF0
XEP=VF1
CLD
GFN
AUTIT.FCP
FRN
GFN
I.000[,IRDATA]
EXT=DFN
DUM: AFP OPERATES ON DFN.
AFP
FPD
OFN=DFN
MOS
RAD
ABD
ASD
END

!SA4
DUM. "SA4" < SAM,STI
RFN=DFN-1
DSR
TEM=9
NPR
DSD
END

!SA5
DUM. "SA5" < SAM,STI
VF0=4000
OMD
INITIAL & FINAL CM-1 FOR 1ST DISPLAY:
VF0
VF1
OMD
INITIAL & FINAL CM-1 FOR 2ND DISPLAY:
VF2
VF3
END

!SA6
DUM. M50-M52 < "SA6" < SAM,STI
SFN=2
OMD

```

*
OMD
1ST SPECTRAL FILE # (FILE DFN-1 MUST HAVE VALID FSB):
DFN
SRT=DFN
OMD
INPUT OPTION: 50 = KEYBOARD INPUT
OMD
                    51 = AUTOSCAN WITH DEFAULT PARAMETERS FROM M51
OMD
                    52 = MANUAL SCAN WITH DEFAULT PARAMETERS FROM M52
CMP=51
CMP
TEM=100
NPR
CMR
OMD
GENERIC FILE TITLE (SEE "ATG" MACRO FOR SPECIAL TITLE OPTIONS):
IMD
OMD
HIT "RETURN" TO START
PAU
END

```

```

!SA9
DUM: "SA9" < SAM & STI
DUM: ROUTING DOCUMENTATION FOR "SAM" MACRO:
DUM: SAM/STI      SET UP INPUT
DUM: SA2-4       COLLECT & DISPLAY SPECTRA
DUM: M12-13      MANUAL DELAY
DUM: M14         DUMMY CONDITIONAL MACRO
DUM: M15         AUTO   DELAY
DUM: M23         MANUAL TITLE
DUM: M50-SA5    KEYBOARD INPUT
DUM: M51 OR 52  DEFAULT   "
END

```

```

!SBS
DUM. "SBS" > SB1,SB0
OMD
"SBS" REGENERATES 1-BEAM FILE FROM ABSORBANCE & 2ND 1-BEAM.
TEM=100
NPR
PFN=49
SPF
SBO
SB1. CONVERT INPUT 1-BEAM (RFN) TO "AUTO-ABSORBANCE" (DFN).
DUM. SET DFN = OUTPUT FILE
DFN=1-CMP

```

```

DFN=DFN*SFN
DFN=CMP*BFN+DFN
MOD
DUM. SAMPLE "AUTO-ABS" = ABS + BG "AUTO-ABS"
OFN=1
FCO=1
FCD=-2*CMP+1
ADD
DUM. SET NSD = # SCANS FOR OUTPUT FILE
PAD
VI1=1-CMP
NSD=VI1*NSD
NSD=CMP*NSB+NSD
DUM. CONVERT SAMPLE TO %T, & RECLASSIFY AS 1-BEAM.
TXD
DUM. "FRN" RECLASSIFIES %T AS 1-BEAM & RESETS FCD.
FRN
OFN=DFN
VAL
FCD=FCD/FCO
MUD
ASD
DSD
RPF
END

!SB0
DUM. "SB0" < SBS
OMD
SAMPLE (SFN), BG (BFN), & ABSORBANCE (OFN) FILES:
SFN
BFN
OFN
OMD
REGENERATE SFN (CMP=0) OR BFN (CMP=1)?
CMP=0
CMP
OMD
CM-1 RANGE FOR SCREEN DISPLAY:
XSP
XEP
DUM. SET RFN = 1-BEAM INPUT FILE:
RFN=1-CMP
RFN=RFN*BFN
RFN=CMP*SFN+RFN
DUM. READ DATA VALUES AT XSP FOR BOTH INPUT FILES FOR CALC OF
DUM. OUTPUT FILE MULTIPLIER (VIA "SBFTYP.FCP").
VAL
VF0=FCO
VI2=OFN
OFN=RFN

```

VAL
VF1=FCO
OFN=VI2
DFN=1
END

!SB1
DUM. "SB1" < SBS & AOB
DUM. CONVERT 1-BEAM FILE (RFN) TO "ABSORBANCE" (DFN) USING
DUM. THE MAX INTENSITY BETWEEN XSP & XEP AS THE "BACKGROUND"
DUM. FOR RFN.
RTN=DFN
RTO=BFN
BFN=RFN
DFN=0
MRD
RAD
DUM. FILE 0 IS NOW 100%T. USE "SBFTYP.FCP" TO RECLASSIFY FILE 0
DUM. AS 1-BEAM WITH: "INTENSITY" = 1/NSR
GFN
SBFTYP.FCP
FRN
PAR
ASR
FCD=NSR*YEP
MUD
DUM. FILE 0 NOW EQUALS YEP FOR ALL CM-1; I.E. NSR*YEP*1/NSR=YEP
DFN=RTN
DUM. RATIO RFN TO RMAX (YEP CANCELS OUT)
MRD
BFN=0
RAD
ABD
BFN=RTO
END

!SMH
OMD
"SMH": ***** SPECTRUM SMOOTHING MACRO *****
OMD
INPUT (OFN) & OUTPUT (DFN) FILES:
OFN
DFN
OMD
OF POINTS:
PTS
OMD
CM-1 RANGE:
XSP

XEP
OMD
DYNAMIC (YS) OR NORMAL (NO) SMOOTHING?
SMO
OMD
MINIMUM NOISE (%T):
THR
VIO=0
GFN
SMOOTH.FTN
FRN
END

!SMT
OMD
** "SMT" SMOOTHS A SPECTRUM & WRITES OUTPUT TO A NEW FILE **
OMD
FILE TO BE SMOOTHED:
SFN
PAS
OMD
OUTPUT FILE:
DFN
OMD
HALF-WIDTH AT HALF MAXIMUM (CM-1 OR WVL) OF SMOOTHING FUNCTION:
VFO
OMD
X-AXIS RANGE TO BE SMOOTHED:
XSP
XEP
MSD
TEM=4
NPR
GFN
SMT.FCP
FRN
ASD
DSD
NPR
END

!STI
DUM. "STI" > SA(0,2-6,9),M(12-15,17,23,50-52)
OMD
"STI" COLLECTS & DISPLAYS SAMPLE SPECTRA; IT PAUSES FOR KEYBOARD
OMD
INPUT OF A TITLE PRIOR TO EACH DATA COLLECTION.
OMD
*

SA6
RTN=1
SA2
END

!STK
OMD
"STK" GENERATES A STICK SPECTRUM FROM ABSORBANCE DATA IN NICOS
OMD
ASCII FILE "STICK.DAT". SEE DIRECTIONS IN "STICK.FOR" FOR SETUP
OMD
OF "STICK.DAT".
OMD
*
OMD
SPECIFY A JUNK SCRATCH FILE THAT ALREADY CONTAINS ABSORBANCE
OMD
DATA. THE STICK SPECTRUM WILL BE SENT TO THIS FILE.
DFN
TXD
OMD
ENTER DATA SET # OF NICOS FILE "STICK.DAT".
VIO
PAD
GFN
STICK.FCP
FRN
ABD
ASD
DSD
END

!SUM
OMD
*** "SUM" ADDS RFN TO DFN AS FOLLOWS:
OMD
*
OMD
FCR * RFN + FCD * DFN + VF3 = OFN
OMD
*
OMD
VF3 IS AN ADDITIVE CONSTANT; OFN IS THE OUTPUT FILE
OMD

OMD
CM-1 RANGE TO BE ADDED:
FXF
LXF
XSP=FXF
XEP=LXF

OMD
INPUT (RFN, DFN) & OUTPUT (OFN) FILE #'S:
RFN
DFN
OFN
OMD
SCALE FACTORS:
FCR
FCD
OMD
ADDITIVE CONSTANT:
VF3=0
VF3
PAD
SU0. CALL "SUMSPC.FCP"
SFN=OFN
ASS
DSS
END

!SU0
DUM. "SU0" < SUM
TEM=100
NPR
ASR
VF1=YEP
ASD
VF2=YEP
GFN
SUMSPC.FCP
FRN
END

!SVI
DUM: "SVI" > SV0-SV3
OMD
*** INTERACTIVE SUBTRACTION OF H2O & CO2 VAPOR ***
SV2
GFN
H2OA2.000[VAPOR
SV3
XSP=1700
XEP=1600
SV0
ASS
DSS
FCR=1
OMD
SET "FCR" NEGATIVE IF VAPOR PEAKS ARE NEGATIVE ON GREEN DISPLAY:
FCR

```

SV1
GFN
H2O A13.000[VAPOR
SV3
XSP=3800
XEP=3600
SV0
SV1
GFN
CO2 A3.000[VAPOR
SV3
XSP=2400
XEP=2280
SV0
ASS
DSS
FCR=1
OMD
SET "FCR" NEGATIVE IF VAPOR PEAKS ARE NEGATIVE ON GREEN DISPLAY:
FCR
SV1
GFN
CO2 A2.000[VAPOR
SV3
XSP=625
XEP=715
SV0
SV1
GFN
CO2 A1.000[VAPOR
SV3
XSP=3760
XEP=3550
SV0
SV1
RPF
END

!SVP
DUM: "SVP" > SV0-SV3
OMD
*** INTERACTIVE SUBTRACTION OF H2O & CO2 VAPOR ***
OMD
"SVP" USES THE SAME MULTIPLIER FOR ALL BANDS OF A GIVEN VAPOR.
OMD
USE "SVI" IF INDIVIDUAL MULTIPLIERS ARE DESIRED.
SV2
GFN
WAT123.000[VAPOR
SV3
XSP=1700
XEP=1600
SV0

```

```
ASS
DSS
FCR=1
OMD
SET "FCR" NEGATIVE IF VAPOR PEAKS ARE NEGATIVE ON GREEN DISPLAY:
FCR
SV1
GFN
CD123.000[VAPOR
SV3
XSP=2400
XEP=2280
SV0
ASS
DSS
FCR=1
OMD
SET "FCR" NEGATIVE IF VAPOR PEAKS ARE NEGATIVE ON GREEN DISPLAY:
FCR
SV1
RPF
END
```

```
!SV0
DUM: "SV0" < SVI & SVP
DUM: DISPLAY VAPOR SPECTRUM & FREEZE IT
ASR
DSR
TEM=9
NPR
END
```

```
!SV1
DUM: "SV1" < SVI & SVP
OMD
*
OMD
1) DO INTERACTIVE SUBTRACTION USING X-ZOOM KEYS
OMD
2) USE <CNTRL E> TO SAVE SUBTRACTION; OTHERWISE GO TO STEP #3
OMD
3) USE <CR> TO RESUME MACRO:
SUB
PAU
TEM=5
NPR
END
```

!SV2
DUM: "SV2" < SVI & SVP
OMD
*
OMD
REFERENCE VAPOR SPECTRA ARE ASSUMED TO BE IN DIRECTORY [VAPOR].
OMD
THE FILE EXTENSIONS DENOTE RESOLUTION (E.G. EXT=002 FOR 2 CM-1).
OMD
*
DUM: SYSTEM MACRO "SUB" SUBTRACTS RFN FROM SFN; "SV2" PUTS
DUM: RFN=1, & SETS DFN=RFN SO THAT "AFG" CAN RETRIEVE THE
DUM: 1ST VAPOR SPECTRUM INTO FILE 1.
OMD
FILE # FOR VAPOR-CONTAMINATED ABSORBANCE SPECTRUM:
SFN
PFN=49
SPF
PAS
RFN=1
DFN=1
FCR=1
END

!SV3
DUM: "SV3" < SVI & SVP
DUM: EXT=0 FOR 0.5 CM-1 RES; 2 FOR 2 CM-1.
EXT=32769/NTP-1
AFG
END

!TTL
OMD
*** "TTL" PRINTS A BLOCK OF FTIR FILE TITLES & START TIMES ***
OMD
1ST FILE #:
VIO
OMD
LAST FILE #:
QIT
DFN=VIO
DUM: USE SUB-BLOCKS OF SIZE RTN FOR SCREEN DISPLAY:
RTN=8
RTO=1
VI1=QIT-VIO+1
VI2=VI1
VI3=VI2/RTN
FOR III=1 TIL VI3

DUM: CALC UPPER INDEX (RTQ) FOR SUB-BLOCK:

RTO=RTO+1
RTP=RTO/VI3
RTP=2-RTP
RTQ=1-RTP
RTP=RTN*RTP
RTQ=VI2*RTQ+RTP
FOR JJJ=1 TIL RTQ
OMD

PRN DFN
TIQ
STD
DFN=DFN+1
NXT JJJ
VI2=VI2-RTN
OMD

PAU/ TYPE [CR] TO CONTINUE
NXT III
END

!UVV

OMD
*** "UVV" COPIES UV-VIS DATA FROM A USER-SPECIFIED NICOS FILE TO
OMD
A BLOCK OF FTIR FILES. THE UV-VIS DATA ARE OBTAINED BY INTER-
OMD
COMPUTER TRANSFER: SND.OY (PE-7500) > PDAPAK.FOR (ZENITH) >
OMD
XMODEM (NICOLET).
OMD
*
OMD
OF FILES:
QIT
OMD
1ST FILE:
DFN
OMD
NICOS FILE NAME & DIRECTORY:
TID
GFN
UVVIS.FCP
FRN
XSP=190
XEP=900
ASD
DSD
END

```

C -----
C "AMAC.FOR"
C
C CALLS: "SORT.FOR"
C -----
C
PROGRAM AMAC
INTEGER*2 ISIZE
INTEGER ISORT(200),NAM(3)
REAL CODE(200)
CHARACTER CHR(64)*1, BLANK*1,GEN*6,EX(10)*2,FILNAM*8,
$      CONC*64,EOF*2,CHRS*64,TRIM*64
DATA BLANK/' '/,
$      EX/'0','1','2','3','4','5','6','7','8','9'/
$      ,GEN/'JUNK.'/,
$      EOF/'!!!'/

C
OPEN(10,FILE='MACROSAVE',FORM='FORMATTED',STATUS='OLD')
C
WRITE(2,600)
WRITE(2,700)
WRITE(2,800)
WRITE(2,900)
C
C INITIALIZE MACRO COUNTER "M" & LINE COUNTER "L":
C
M = 0
L = 1
C
C LOOK FOR AN EXCLAMATION MARK DENOTING THE BEGINNING OF A NEW
C MACRO.
C
11 READ(10,100,END=101) CHR
IF (ICHAR(CHR(1)).NE.33) GO TO 11
C
C INCREMENT THE EXTENSION # OF THE GENERIC OUTPUT JUNK FILES,
C & GENERATE THE CHARACTER VARIABLE "FILNAM".
C
21 M = M+1
N1 = M/100 + 1
N2 = ( M - 100*(N1-1) )/10 + 1
N3 = M - 100*(N1-1) - 10*(N2-1) + 1
FILNAM = CONC ( GEN, EX(N1), EX(N2), EX(N3) )
OPEN(12,FILE= FILNAM ,FORM='FORMATTED',STATUS='NEW')
WRITE(12,100) CHR
C
C ASSIGN AN ALPHABETIZING CODE TO THE MACRO TITLE, WITH DIGITS
C RANKED BELOW LETTERS:
C
DO 31 I = 1, 3
NAM(I) = ICHAR( CHR( I + 1 ) )
31 IF ( NAM(I).LT. 65 ) NAM(I) = NAM(I) + 43
CODE(M) = 40000.*NAM(1) + 200.*NAM(2) + NAM(3)

```

```

        WRITE(2,300) CHR(2), CHR(3), CHR(4)
C
C CONTINUE READING THE FILE UNTIL THE BEGINNING OF THE NEXT
C MACRO IS ENCOUNTERED, OR 2 SUCCESSIVE EXCLAMATION MARKS ARE
C FOUND.
C
41    READ(10,100,END=101) CHR
      L = L + 1
      IF (ICHAR(CHR(1)).EQ.33) CLOSE(12)
      IF (ICHAR(CHR(1)).EQ.33.AND.ICHAR(CHR(2)).NE.33 ) GOTO 21
      IF (ICHAR(CHR(1)).EQ.33.AND.ICHAR(CHR(2)).EQ.33 ) GOTO 111
      WRITE(12,100) CHR
                                                    GOTO 41

C
C *****
C SORT THE ALPHABETIZING CODES:
C
101   CLOSE(12)
      PRINT *,'ERROR: EOF W/O TWO EXCLAMATION MARKS.'
111   CALL SORT( M, CODE, ISORT )
C
C OPEN THE OUTPUT FILE:
C
      OPEN(11, FILE = 'AMAC.NEW', FORM='FORMATTED',
$       STATUS='UNKNOWN', SIZE=ISIZE)
C
C RETRIEVE THE MACROS IN ALPHABETICAL ORDER FROM THE "JUNK.EXT"
C FILES. INSERT A BLANK LINE BETWEEN EACH PAIR OF SUCCESSIVE
C MACROS. TRIM TRAILING BLANKS FROM OUTPUT.
C
      WRITE(2,400)
DO 131 I = 1, M
      M1 = CODE(I)/40000
      M2 = (CODE(I) - M1*40000.)/200
      M3 = CODE(I) - M1*40000. - M2*200.
      IF ( M1. GT. 90 ) M1 = M1 - 43
      IF ( M2. GT. 90 ) M2 = M2 - 43
      IF ( M3. GT. 90 ) M3 = M3 - 43
      WRITE(2,300) CHAR(M1), CHAR(M2), CHAR(M3)
      IEXT = ISORT(I)
      N1 = IEXT/100 + 1
      N2 = ( IEXT - 100*(N1-1) )/10 + 1
      N3 = IEXT - 100*(N1-1) - 10*(N2-1) + 1
      FILNAM = CONC ( GEN, EX(N1), EX(N2), EX(N3) )
      OPEN(12,FILE= FILNAM,FORM='FORMATTED',STATUS='OLD')
DO 121 L = 1, 1000
      READ(12,200,END=125) CHRS
121   WRITE(11,200) TRIM(CHRS,64)
125   WRITE(11,100) BLANK
131   CLOSE(12)
      WRITE(11,500) EOF
100   FORMAT(64A1)
200   FORMAT(A64)

```



```

300  FORMAT(1X,3A1)
400  FORMAT('0','ALPHABETIZED MACROS')
500  FORMAT(A2)
600  FORMAT(1X,'THIS PROGRAM ALPHABETIZES AN ASCII FILE CALLED
$ "MACROSAVE"')
700  FORMAT(1X,'(LIBRARY OF MACROS) & SENDS THE OUTPUT TO
$ "AMAC.NEW"')
800  FORMAT('0','***** WARNING *****')
900  FORMAT(1X,'THIS PROGRAM WILL FAIL IF THE EOF SYMBOL IN
$ MACROSAVE IS ON THE SAME LINE AS THE !! MARKER')
      CLOSE(11)
      STOP
      END

```

```

!
C -----
C "ASCSND.FOR"
C
C CALLS: 1) "FTPARM.FOR"
C         2) FTIR SYSTEM SUBROUTINES
C
C CALLING MACRO: ASC
C -----
C
C      PROGRAM ASCSND
C
C THIS PROGRAM SENDS A BLOCK OF FTIR SCRATCH FILES TO ANOTHER
C COMPUTER VIA THE PRINTER PORT. IT IS CALLED BY MACRO "ASC".
C
C THE DIMENSION 3584 CORRESPONDS TO THE # OF POINTS IN ONE
C SECTOR OF 7 COLUMNS OF SPECTRAL DATA SENT TO DISK.
C
C      INTEGER IDATA(512),FSBSEC
C      DIMENSION Y(3584),DSCALE(7)
C
C READ FTIR PARAMETERS:
C
C      INODE  =IRVAL(13004,0)
C      NSECS  =IRVAL(14001,0)
C      IFILE  =IRVAL(14022,0)
C      WVNI   = RVAL(14135,0)
C      WVNL   = RVAL(14213,0)
C      NFILES =IRVAL(14222,0)
C
C READ THE FILE STATUS BLOCKS TO GET THE HENE LASER CM-1 & THE
C FILE EXPONENTS.
C
C      FSBSEC = IFILE*NSECS+88-1
C      DO 11 I=1,NFILES

```

```

CALL IRTISK(IDATA,512,FSBSEC+I*NSECS,INODE)
IEXP = IDATA(6)
DSCALE(I) = 2**(19-IEXP)
IF (I.EQ.1) THEN
    IHENE = IDATA(333)
    CALL FTPARM(IHENE,WVNI,WVNL,WVNMIN,WVNMAX,
                IISEC,IFSEC,INDXSP,INDXEP,PPWVN)
$
    ENDIF
11
C
CONTINUE
DO 201 M = IISEC,IFSEC
    WRDDIF = 512*(M-1)
    L = 0
    JMIN = 1
    JMAX = 512
    IF (M.EQ.IISEC) JMIN = INDXSP
    IF (M.EQ.IFSEC) JMAX = INDXEP
    NPPSEC = JMAX-JMIN+1
DO 101 I = 1,NFILES
    MSEC=FSBSEC+(I-1)*NSECS+M
    CALL IRTISK(IDATA,512,MSEC,INODE)
DO 31 J = JMIN,JMAX
    L = L+1
    Y(L) = IDATA(J)/DSCALE(I)
    IF (Y(L).GT.4.0) Y(L) = 4.0
    IF (Y(L).LT.-.99) Y(L) = -.99
31
101
CONTINUE
N1 = NPPSEC
N2 = 2*NPPSEC
N3 = 3*NPPSEC
N4 = 4*NPPSEC
N5 = 5*NPPSEC
N6 = 6*NPPSEC
L = 0
DO 111 J = JMIN,JMAX
    L = L+1
    WVN = (J+WRDDIF-1)/PPWVN
    IF (NFILES.EQ.1) THEN
        WRITE(10,300) WVN,Y(L)
    ELSEIF (NFILES.EQ.2) THEN
        WRITE(10,300) WVN,Y(L),Y(L+N1)
    ELSEIF (NFILES.EQ.3) THEN
        WRITE(10,300) WVN,Y(L),Y(L+N1),Y(L+N2)
    ELSEIF (NFILES.EQ.4) THEN
        WRITE(10,300) WVN,Y(L),Y(L+N1),Y(L+N2),
$
        Y(L+N3)
    ELSEIF (NFILES.EQ.5) THEN
        WRITE(10,300) WVN,Y(L),Y(L+N1),Y(L+N2),
$
        Y(L+N3),Y(L+N4)
    ELSEIF (NFILES.EQ.6) THEN
        WRITE(10,300) WVN,Y(L),Y(L+N1),Y(L+N2),
$
        Y(L+N3),Y(L+N4),Y(L+N5)
    ELSE

```

```

                WRITE(10,300) WVN,Y(L),Y(L+N1),Y(L+N2),
$                Y(L+N3),Y(L+N4),Y(L+N5),Y(L+N6)
                ENDIF
111            CONTINUE
201            CONTINUE
                WRITE(10,400)
300            FORMAT(1X,F6.1,7(1X,F6.4))
400            FORMAT(1X,' ')
                CALL EXIT
                END

```

```

!
C -----
C "AUTIT.FOR"
C
C CALLS: 1) "AUTIT1.FOR"
C         2) FTIR SYSTEM SUBROUTINES
C
C CALLING MACROS: LBB, SAM, & STI
C -----
C
C PROGRAM AUTIT
C
C AUTOMATIC TITLE GENERATOR FOR FTIR FILES.
C
C THIS PROGRAM SHOULD BE CALLED BY A SPECTRAL DATA
C COLLECTION MACRO EACH TIME A SPECTRUM IS COLLECTED, AND
C PREFERABLY BEFORE THE SPECTRUM IS DISPLAYED. THE "SRT"
C PARAMETER IS USED TO STORE THE FILE # OF THE 1ST FILE
C IN THE BLOCK OF SPECTRA SO THAT FILE EXTENSIONS CAN BE
C ASSIGNED CORRECTLY, & THE TITLE SPECIFICATIONS CAN BE
C RETRIEVED FROM FILE SRT-1 (NOTE: TITLE SPECS CANNOT BE STORED
C IN FILES SRT OR HIGHER SINCE THE COMPUTER WRITES A NEW FSB
C EACH TIME IT COLLECTS DATA). "DFN" IS THE CURRENT FILE TO BE
C TITLED.
C
C INTEGER IFSB(512),OFSB(512),OFILE,LDIG(4),SRTFIL
C
C READ FTIR PARAMETERS; THEN READ THE STATUS BLOCK OF THE FILE
C WHICH CONTAINS THE KEYBOARD INPUT SPECIFICATIONS FOR THE
C TITLES OF THE SPECTRAL FILES:
C
C FTIR VS. FORTRAN:
C         SRT = SRTFIL
C         DFN = OFILE
C
C IQUIT = IRVAL(13736,0)
C IF (IQUIT.EQ.1)

```

GO TO 701

```

C
  INODE = IRVAL(13004,0)
  NDP   = IRVAL(14000,0)*256
  NSECS = IRVAL(14001,0)
  OFILE = IRVAL(14022,0)
  SRTFIL= IRVAL(14221,0)

C
  CALL IRTISK(IFSB,512,SRTFIL*NSECS+87,INODE)

C
C READ THE OUTPUT STATUS BLOCK ARRAY WHICH HAS BEEN CREATED BY
C THE FTIR SYSTEM AFTER SPECTRAL DATA COLLECTION:
C
  CALL IRTISK(OFSB,512,(OFILE+1)*NSECS+87,INODE)

C
C INITIALIZE THE TITLE FIELD OF "OFILE" WITH BLANKS:
C
  DO 11 I=68,107
11      OFSB(I)=160
C *****
C
C TITLE FIELD #1:  GENERIC FILE NAME
C
  NCI= 144
  NCO= 67
  CALL AUTIT1(41,NCI,NCO,IFSB,OFSB)
  IF (NCO.GE.107)                                GO TO 601

C
C IF FIELD #1 OR #2 IS EMPTY,  SKIP TO FIELD #3.  OTHERWISE, PUT
C A PERIOD AFTER THE GENERIC FILE NAME:
C
  NCI=NCI+1
  IF (NCO.EQ.67)                                GO TO 301
  NCO=NCO+1
  IF (IFSB(NCI).EQ.170)                          GO TO 221
  IF (NCO.LE.107) OFSB(NCO)=174

C
C *****
C
C TITLE FIELD #2:  FILE EXTENSION #
C
  NDIG=0
  DO 201 I=1,4
  IF (IFSB(NCI).EQ.170)                          GO TO 211
  NCI=NCI+1
201  NDIG=NDIG+1
  PRINT *,'ERROR: NO * AFTER TITLE EXTENSION FIELD'
  GO TO 601
211  IEXT=OFILE-SRTFIL + IFSB(NCI-1)-176
  IF (NDIG.GT.1) IEXT=IEXT+ 10*(IFSB(NCI-2)-176)
  IF (NDIG.GT.2) IEXT=IEXT+100*(IFSB(NCI-3)-176)
  IDIG=IEXT/100
  OFSB(NCO+1)=IDIG+176
  IREM=IEXT-100*IDIG

```

```
IDIG=IREM/10
OFSB(NCO+2)=IDIG+176
OFSB(NCO+3)=IREM-10*IDIG+176
```

```
C
C PUT A BLANK AFTER THE FILE EXTENSION #:
C
C NCO=NCO+4
221 IF (NCO.LE.107) OFSB(NCO)=160
C
C *****
C
C TITLE FIELD #3: DESCRIPTOR
C
C EXAMINE THE 1ST CHARACTER OF THE DESCRIPTOR. IF IT IS AN
C ASTERISK, SKIP TO THE NEXT FIELD; IF IT IS A BLANK, IGNORE IT
C SINCE A BLANK HAS ALREADY BEEN PLACED AFTER THE FILE
C EXTENSION #. IF IT IS NOT A BLANK, SEND IT TO THE OUTPUT FSB
C ARRAY.
C
301 NCI=NCI+1
    IF (IFSB(NCI).EQ.170) GO TO 401
    IF (IFSB(NCI).EQ.160) GO TO 311
    NCO=NCO+1
    IF (NCO.LE.107) OFSB(NCO)=IFSB(NCI)
311 CALL AUTIT1(41,NCI,NCO,IFSB,OFSB)
C
C IF THE LAST LETTER OF THE DESCRIPTOR WAS A BLANK, SKIP TO
C FIELD #4. IF IT WASN'T, INSERT A BLANK INTO THE OUTPUT ARRAY.
C
    IF (IFSB(NCI-1).EQ.160) GO TO 401
    NCO=NCO+1
    OFSB(NCO)=160
C
C *****
C TITLE FIELD #4: ELAPSED TIME
C
C DETERMINE WHETHER THE MONTH OF THE CURRENT SPECTRUM AGREES
C WITH THE MONTH OF THE T=0 TIME. ASSUME THAT THE MONTH HAS
C NOT CHANGED BY MORE THAN 1 UNIT:
C
401 NCI = NCI + 1
    IF (IFSB(NCI).EQ.170) GO TO 501
    MONDIF= OFSB(50) - IFSB(NCI+1)
    INCDAY=0
    IF (MONDIF.EQ.0) GO TO 411
C
C COMPUTE TIME DIFFERENCES DIRECTLY FROM ASCII INTEGER CODES.
C COMPUTE ABSOLUTE TIMES BY SUBTRACTING 176 FROM ASCII CODES.
C
    MO= 10*(IFSB(NCI)-176) + IFSB(NCI+1) - 176
    IF (MO.EQ.4.OR.MO.EQ.6.OR.MO.EQ.9.OR.MO.EQ.11) INCDAY=30
    IF (MO.EQ.2) INCDAY=28
```

```

411 DAYDIF= 10 * (OFSB(52)-IFSB(NCI+2))
DAYDIF= DAYDIF+ OFSB(53) - IFSB(NCI+3) + INCDAY
HRDIF = 10 * (OFSB(58)-IFSB(NCI+4))
HRDIF = HRDIF + OFSB(59) - IFSB(NCI+5) + DAYDIF*24
FHRDIF= 10 * (OFSB(61)-IFSB(NCI+6))
FHRDIF= (FHRDIF + OFSB(62) - IFSB(NCI+7) ) / 60
HRDIF = HRDIF + FHRDIF

C
C DETERMINE EACH DIGIT OF THE ELAPSED TIME, CONVERT EACH BACK TO
C ASCII CODE, AND SEND CODES TO THE OUTPUT FSB:
C
IF (HRDIF.LT.0.0) PRINT *, 'ERROR: NEGATIVE ELAPSED TIME'
IF (HRDIF.LT.0.0) GO TO 601
NCO = NCO + 1
421 N = ALOG10(HRDIF)
OFSB(NCO) = 176
REM = HRDIF
DO 431 I = 1, 4
J = I - 1
IF (I. GT. N+1) GO TO 441
Q = 10**(N-J)
LDIG(I) = INT( 0.001 + REM/Q )
REM = REM - Q*LDIG(I)
431 OFSB(NCO+J) = LDIG(I) + 176
441 IF (N.LT.0) J=1
C
C SEND ".# HR" TO THE OUTPUT ARRAY:
C
OFSB(NCO+J) = 174
NR = NINT(10*REM)
IF (NR. EQ. 10) HRDIF = HRDIF + 0.05
IF (NR. EQ. 10) GO TO 421
OFSB(NCO+J+1) = NR + 176
OFSB(NCO+J+2) = 160
OFSB(NCO+J+3) = 200
OFSB(NCO+J+4) = 210
OFSB(NCO+J+5) = 160
NCO = NCO + J + 5
NCI = NCI + 8

C
C *****
C TITLE FIELD #5: DATE
C
501 NCI = NCI + 1
C
C OMIT THE DATE IF NO "D" APPEARS IN INPUT TITLE SPECIFICATION
C FIELD #5 OR IF THERE IS NOT ENOUGH SPACE LEFT IN THE OUTPUT
C TITLE FIELD:
C
IF (IFSB(NCI).NE.196) GO TO 601
IF (NCO.GT.100) GO TO 601
CALL AUTIT1(8,48,NCO,OFSB,OFSB)
601 CALL IWTISK(OFSB,512,(OFILE+1)*NSECS+87,INODE)

```

701 CALL EXIT
END

```
!  
C -----  
C "AUTIT1.FOR"  
C  
C SUBROUTINE OF "AUTIT.FOR"  
C -----  
C  
C SUBROUTINE AUTIT1(IMAX,NCI,NCO,IFSB,OFSB)  
C  
C INTEGER IFSB(512),OFSB(512)  
C  
C DIRECTLY TRANSFER THE CHARACTERS OF THE INPUT TITLE FIELD TO  
C THE OUTPUT TITLE FIELD UNTIL AN ASTERISK (ASCII CODE 170) IS  
C ENCOUNTERED OR "IMAX" CHARACTERS HAVE BEEN TRANSFERRED.  
C  
C DO 101 I=1,IMAX  
C NCI=NCI+1  
C IF (IFSB(NCI).EQ.170) GO TO 201  
C NCO=NCO+1  
101 IF (NCO.LE.107) OFSB(NCO)=IFSB(NCI)  
201 RETURN  
C END
```

```
!  
C -----  
C "AXIS2.FOR"  
C  
C SUBROUTINE OF "PLOTFT.FOR"  
C -----  
C  
C  
C SUBROUTINE AXIS2 (USTART, UEND, XPNO, YPNO, AXSPAN,  
1 AXANGL, SZMJTC, TICRAT, TSZRAT, BEGNUM,  
2 SPCNUM, DECDIG, SIZNUM, DSTNUM, JSTNUM,  
3 ANGNUM, LETTRS, NLET , SIZLET, DSTLET,  
4 CNTLET, ANGLET)  
C  
C INTEGER TICRAT,TR1,DECDIG,AXFLIP,SGNAXS,LETTRS(88)  
C REAL NUMJST,JSTNUM  
C  
C *****  
C A NEGATIVE "SIZNUM" KILLS TICS & TIC LABELS.
```

```

C " " "SIZLET" " AXIS LABEL.
C
IF (DECDIG.LE.0) DECDIG=-1
SPANMX=0.0
IAXROT=0
NUMROT=0
LETROT=0
AXFLIP=0
NUMFLP=0
LETFLP=0
SGNAXS=1
NSTEPS=0
ILABEL=0
SPACNM=SPCNUM
DISLET=DSTLET
NUMJST=SIZNUM*JSTNUM
IF (NINT(AXANGL).EQ.90.OR.NINT(AXANGL).EQ.270) IAXROT=1
IF (NINT(AXANGL).EQ.180.OR.NINT(AXANGL).EQ.270) AXFLIP=1
IF (NINT(ANGNUM).EQ.90.OR.NINT(ANGNUM).EQ.270) NUMROT=1
IF (NINT(ANGNUM).EQ.180.OR.NINT(ANGNUM).EQ.270) NUMFLP=1
IF (NINT(ANGLET).EQ.90.OR.NINT(ANGLET).EQ.270) LETROT=1
IF (NINT(ANGLET).EQ.180.OR.NINT(ANGLET).EQ.270) LETFLP=1
S1=2*AXFLIP-1
S2=2*NUMFLP-1
S3=NUMFLP-AXFLIP
S4=1-AXFLIP-NUMFLP
S5=2*LETFLP-1
S6=LETFLP-AXFLIP
S7=1-AXFLIP-LETFLP
C
C *****
C ***** DRAW AXIS *****
C
IF (AXSPAN.LT.0) SGNAXS=-1
C
C INITIALIZE PLOTTER AND THEN MOVE PEN TO AXIS ORIGIN.
C
CALL PLOT(XPNO,YPNO,3)
C
C LOCATE END OF AXIS; THEN DRAW AND RETRACE THE AXIS.
C
XPN=XPNO+(1-IAXROT)*AXSPAN
YPN=YPNO+IAXROT*AXSPAN
CALL PLOT(XPN,YPN,2)
CALL PLOT(XPNO,YPNO,2)
IF (SIZNUM.LE.0.0) GO TO 121
C
C DETERMINE TIC LOCATION IN INCHES (TICLOC) FOR FIRST TIC:
C
IF (UEND.LT.USTART.AND.SPACNM.GT.0.0) SPACNM=-SPACNM
UPINCH=(UEND-USTART)/ABS(AXSPAN)
TICLOC=(BEGNUM-USTART)/UPINCH
SPMJTC=SPACNM/UPINCH

```



```

        TR1=TICRAT+1
        SPMINR=SPMJTC/TR1
DO 61 I=1,TICRAT+1
        TICLOC=TICLOC-SPMINR
        IF (TICLOC.LE.-0.01)                GO TO 81
        NSTEPS=NSTEPS+1
61      CONTINUE
81      TICLOC=TICLOC+SPMINR
C
C COMPUTE THE TOTAL NUMBER OF TICS.  A '1.01' VALUE IS USED IN
C PLACE OF '1.00' TO AVOID TRUNCATION ERRORS IN CONVERTING FROM
C A REAL NUMBER TO AN INTEGER.
C
        NTICS=1.01+(ABS(AXSPAN)-TICLOC)/SPMINR
C
C *****
C ***** DRAW AND LABEL THE TIC MARKS *****
C
        ICOUNT=0
        DO 111 I=0,NTICS-1
C
C DETERMINE WHETHER THE UPCOMING TIC IS MAJOR OR MINOR:
C
        ILARGE=0
        NQ=I-NSTEPS
        IF (TR1*(NQ/TR1).EQ.NQ) ILARGE=1
        TICSIZ=ILARGE*SZMJTC+(1-ILARGE)*SZMJTC/TSZRAT
C
C LOCATE THE POSITION OF THE UPCOMING TIC AND THEN MOVE THE
C PEN THERE.
C
        TICNXT=TICLOC+I*SPMINR
        XPN=XPN0+(1-IAXROT)*SGNAXS*TICNXT
        YPN=YPN0+IAXROT*SGNAXS*TICNXT
        CALL PLOT(XPN,YPN,2)
C
C DETERMINE THE COORDINATES OF THE END OF THE TIC MARK AND
C THEN MOVE THE PEN THERE.
C
        XPN2=XPN
        YPN2=YPN
        IF (IAXROT.EQ.1) XPN2=XPN2-S1*TICSIZ
        IF (IAXROT.EQ.0) YPN2=YPN2+S1*TICSIZ
        CALL PLOT(XPN2,YPN2,2)
        IF (ILARGE.NE.1)                GO TO 111
C
C IF TIC IS MINOR, SKIP TO END OF TIC LOOP; IF TIC IS MAJOR,
C DETERMINE HOW MANY CHARACTERS ARE PRESENT IN THE NUMBER LABEL
C (INCLUDING THE DECIMAL POINT, IF PRESENT).  THIS VALUE IS
C NEEDED FOR DETERMINING WHERE TO POSITION THE PEN FOR THE
C 'NUMBER' SUBROUTINE.
C 'NNUM' IS INITIALLY SET AT THE MINIMUM TOTAL DIGITS
C POSSIBLE FOR THE GIVEN # OF DIGITS TO THE RIGHT OF THE DECIMAL

```

C PT. IT IS LATER INCREMENTED BY 'MORDIG' IF THE LOGARITHM OF
C THE # LABEL SHOWS THAT MORE DIGITS ARE REQUIRED.

C
NNUM=2+DECDIG
IF (DECDIG.EQ.0) NNUM=1
RNUM=BEGNUM+ILABEL*SPACNM
IF (RNUM.LT.0) NNUM=NNUM+1
NROUND=NINT(RNUM)
ILABEL=ILABEL+1
IF (NROUND.LT.10) GO TO 101

C
C THE '0.0001' TERM BELOW IS USED TO PREVENT TRUNCATION ERRORS.

C
MORDIG=INT(ALOG10(0.0001+NROUND))
NNUM=NNUM+MORDIG

C
C 'SPANUM' IS THE LENGTH OF THE CURRENT # LABEL IN INCHES.
C 'SPANMX' IS THE LENGTH OF THE LONGEST # LABEL, AND IS LATER
C USED FOR CALCULATING WHERE TO POSITION THE AXIS NAME.

C
101 SPANUM=(NNUM-0.5)*SIZNUM
IF (SPANUM.GE.SPANMX) SPANMX=SPANUM

C
C THE FOLLOWING STEPS ARE USED TO LOCATE THE (X,Y) COORDINATES
C OF THE FIRST CHARACTER OF EACH OF THE NUMBER LABELS (XPNUMB
C & YPNUMB). THIS CALCULATION IS BASED ON THE INPUT PARAMETERS
C 'DSTNUM', 'SIZNUM' AND 'JSTNUM'; AND IT TAKES ACCOUNT OF THE
C VARIOUS NUMBER AND AXIS ORIENTATIONS THAT MAY BE REQUESTED.
C COORDINATES THAT DEPEND ON THE # OF DIGITS IN THE # LABEL MUST
C BE CALCULATED FOR EACH TIC LOOP ITERATION SINCE THE # OF
C DIGITS MAY VARY.

C
IF (ICOUNT.GT.0) GO TO 105
ICOUNT=ICOUNT+1
IF (IAXROT.EQ.0.AND.NUMROT.EQ.1) XN=-S2*0.5*SIZNUM
IF (IAXROT.EQ.1.AND.NUMROT.EQ.1) XN=-S1*DSTNUM+S4*SIZNUM
IF (IAXROT.EQ.0.AND.NUMROT.EQ.0) YN=S1*DSTNUM-S4*SIZNUM
IF (IAXROT.EQ.1.AND.NUMROT.EQ.0) YN=S2*0.5*SIZNUM
105 IF (IAXROT.EQ.0.AND.NUMROT.EQ.0) XN=S2*(SPANUM-NUMJST)
IF (IAXROT.EQ.1.AND.NUMROT.EQ.0) XN=-S1*DSTNUM+S3*SPANUM
IF (IAXROT.EQ.0.AND.NUMROT.EQ.1) YN=S1*DSTNUM-S4*SPANUM
IF (IAXROT.EQ.1.AND.NUMROT.EQ.1) YN=S2*(SPANUM-NUMJST)
XPNUMB=XPNUM2+XN
YPNUMB=YPNUM2+YN

C
C LIFT THE PEN AND MOVE IT TO THE LOWER LEFT CORNER OF THE
C FIRST DIGIT OF THE NUMBER LABEL. THEN, DRAW THE NUMBER,
C RETURN TO THE BOTTOM OF THE TIC MARK, PUT THE PEN DOWN, AND
C RETRACE THE TIC MARK.

C
CALL PLOT(XPNUMB,YPNUMB,3)
CALL NUMBER(XPNUMB,YPNUMB,SIZNUM,RNUM,ANGNUM,DECDIG)
CALL PLOT(XPNUM2,YPNUM2,3)

```

111 CALL PLOT(XPN,YPN,2)
C
C RETURN PEN TO AXIS ORIGIN WITH PEN DOWN.
C
CALL PLOT(XPNO,YPNO,2)
C
C *****
C ***** DRAW THE AXIS NAME *****
C
C USE THE PARAMETERS 'DISLET', 'SIZLET', AND 'NLET' TO DETERMINE
C THE COORDINATES OF THE LOWER LEFT CORNER OF THE 1ST CHARACTER
C OF THE AXIS NAME. THIS CALCULATION TAKES ACCOUNT OF THE
C VARIOUS LETTER AND AXIS ORIENTATIONS THAT MAY BE REQUESTED.
C
IF (IAXROT.EQ.NUMROT) BNDRY=ABS(S3)*SIZNUM
IF (IAXROT.EQ.1.AND.NUMROT.EQ.0) BNDRY=ABS(S4)*SPANMX
IF (IAXROT.EQ.0.AND.NUMROT.EQ.1) BNDRY=ABS(S3)*SPANMX
DISLET=DISLET+BNDRY+IAXROT*ABS(XPNO-XPNUMB)
DISLET=DISLET+(1-IAXROT)*ABS(YPNO-YPNUMB)
121 IF (SIZLET.LE.0.0.OR.NLET.LE.0) GO TO 199
SPANLT=(NLET-0.5)*SIZLET
IF (IAXROT.EQ.0.AND.LETROT.EQ.0) XPN=S5*0.5*SPANLT
IF (IAXROT.EQ.0.AND.LETROT.EQ.1) XPN=-S5*0.5*SIZLET
IF (IAXROT.EQ.1.AND.LETROT.EQ.0) XPN=-S1*DISLET+S6*SPANLT
IF (IAXROT.EQ.1.AND.LETROT.EQ.1) XPN=-S1*DISLET+S7*SIZLET
IF (IAXROT.EQ.0.AND.LETROT.EQ.0) YPN=S1*DISLET-S7*SIZLET
IF (IAXROT.EQ.0.AND.LETROT.EQ.1) YPN=S1*DISLET-S7*SPANLT
IF (IAXROT.EQ.1.AND.LETROT.EQ.0) YPN=S5*0.5*SIZLET
IF (IAXROT.EQ.1.AND.LETROT.EQ.1) YPN=S5*0.5*SPANLT
XPN=XPN+XPNO+(1-IAXROT)*(0.5*AXSPAN+CNTLET)
YPN=YPN+YPNO+IAXROT*(0.5*AXSPAN+CNTLET)
CALL SYMBOL(XPN,YPN,-SIZLET,LFTTRS,ANGLET,NLET)
199 CALL PLOT(0,0,3)
RETURN
END

```

```

!
C -----
C "BASLIN.FOR"
C
C CALLS: 1) "FTPARM.FOR"
C         2) "SPLINE.FOR" > "MATINV.FOR"
C         3) FTIR SYSTEM SUBROUTINES
C
C CALLING MACRO: BSL
C
C -----
C
PROGRAM BASLIN

```

C
 C THIS PROGRAM IS INTENDED PRIMARILY FOR REMOVING FRINGES FROM
 C MATRIX-ISOLATION SPECTRA. BASELINE PTS ARE SENT TO FILE 0
 C & NICOS FILE "BASLIN.DAT". POORLY CHOSEN BASELINE PTS CAN BE
 C DELETED FROM "BASLIN.DAT" USING TED. RERUNNING "BSL" USING
 C THE VII=0 OPTION WILL GENERATE A REVISED BASELINE-CORRECTED
 C SPECTRUM. CM-1 RANGES CONTAINING BROAD PEAKS SHOULD BE
 C SKIPPED AS THEY WILL OTHERWISE PRODUCE BAD BASELINE REF PTS.
 C

```

    INTEGER    DRDATA(512),OZDATA(512),NICSEC(4),NICWRD(4),
1             FSKIP(3,2),IBSKIP(3,2),LIMWRD(3,2),
2             BASCNT,BSLINE,BLKSIZ,ENDATA,OZFN,OZFSEC,BSFIND,
3             ODFSEC,ODFN,OLDSEC,ZSCALE
    INTEGER*2  ISIZE
    DIMENSION BFRQ(10),YD(10),CD(10,3)
  
```

C
 C DATA NBLANK/0/ ,LASTPT/0/ ,ICNT /0/ ,ICD /0/ ,
 1 BASCNT/0/ ,ENDATA/512/ ,IH /4/ ,
 2 OLDSEC/0/ ,ISTART/1/

C
 C *****
 C

```

    OPEN(11,FILE='BASLIN.DAT', FORM='FORMATTED',
    $      STATUS='UNKNOWN', SIZE=ISIZE)
  
```

C
 C *****
 C ***** FTIR PARAMETERS *****
 C

```

    BSFIND      = IRVAL(13763,0)
    IRFN        = IRVAL(14030,0)
    ODFN        = IRVAL(14022,0)
    OZFN        = IRVAL(14023,0)
    RFXF        = RVAL(14135,0)
    RLXF        = RVAL(14213,0)
    NSECS       = IRVAL(14001,0)
    INODE       = IRVAL(13004,0)
    FSKIP(1,1)  = IRVAL(13732,0)
    FSKIP(1,2)  = IRVAL(13715,0)
    FSKIP(2,1)  = IRVAL(13736,0)
    FSKIP(2,2)  = IRVAL(14177,0)
    FSKIP(3,1)  = IRVAL(14012,0)
    FSKIP(3,2)  = IRVAL(13760,0)
  
```

C
 C COMPUTE SECTOR INDICES FOR USE IN IRTISK AND IWTISK ROUTINES:
 C

```

    IRFSEC=IRFN*NSECS+88-1
    ODFSEC=ODFN*NSECS+88-1
    OZFSEC=OZFN*NSECS+88-1
  
```

C
 C READ FILE STATUS BLOCK TO GET HE-NE LASER CM-1:
 C

```

    CALL IRTISK(DRDATA,512,IRFSEC+NSECS,INODE)
    IHENE=DRDATA(333)
  
```

```

C
C *****
C COMPUTE SPECTRAL RANGE PARAMETERS WITH "FTPARM.FOR"
C
    CALL FTPARM(IHENE,RFXF,RLXF,FRQMIN,FRQMAX,IISEC,IFSEC,
1          INDXSP,INDXEP,PPWVN)
C
    NX=-524288
    ZSCALE=-NX/4
    IF (BSFIND.EQ.0)                                GO TO 261
    WRITE (2,8000)
    BLKSIZ = IRVAL(13762,0)
    BLKSIZ=16*(BLKSIZ/16)
    IF (BLKSIZ.LT.16) BLKSIZ=16
    NBLKS=512/BLKSIZ
C *****
C COMPUTE ARRAY INDICES CORRESPONDING TO THE CM-1 LIMITS OF
C RANGES TO BE SKIPPED BY THE BASELINE-LOCATOR ALGORITHM:
C
    DO 31 I = 1, 3
        IF ( FSKIP(I,1). GT. FSKIP(I,2) ) THEN
            TMP = FSKIP(I,2)
            FSKIP(I,2) = FSKIP(I,1)
            FSKIP(I,1) = TMP
        ENDIF
31      CONTINUE
    DO 51 I = 1, 2
    DO 41 J = 1, 3
        IX = PPWVN * FSKIP(J,I)
        IBSKIP(J,I) = IX / BLKSIZ + 1
41      LIMWRD(J,I) = IX - ( (IX-1)/512 ) * 512
51      CONTINUE
C
C *****
C ***** LOCATE BASELINE POINTS *****
C
    DO 251 M = IISEC, IFSEC
        IIBLK = 1
        WRDDIF = 512 * (M-1)
        CALL IRTISK( DRDATA, 512, IRFSEC+M, INODE)
        IF ( M. NE. IISEC )                                GO TO 191
C
C *****
C FIRST SECTOR ONLY:
C
        IIBLK = INDXSP / BLKSIZ + 1
        FRQMIN = ( WRDDIF + INDXSP - 1 ) / PPWVN
        WRITE(11,8300) BLKSIZ
        WRITE(11,8500) FRQMIN, RFXF, RLXF
C
C *****
C THE CURRENT SPECTRAL DATA BLOCK IS EXAMINED TO FIND THE
C MINIMUM ABSORBANCE.

```

```

C
191 DO 241 LOCBLK = IIBLK, NBLKS
      IBLK = ( M-1 ) * NBLKS + LOCBLK
      IF ( M. EQ. IFSEC. AND. LOCBLK*BLKSIZ. GE. INDXEP)
1        LASTPT = INDXEP
        ILAST = LOCBLK * BLKSIZ
        IFIRST = ILAST - BLKSIZ + 1
        IF (BASCNT.EQ.0) IFIRST = INDXSP
        IF (LASTPT.NE.0) ILAST = INDXEP
      DO 195 I = 1, 3
        IF (IBLK.GT.IBSKIP(I,1).AND.IBLK.LT.IBSKIP(I,2))
1          GO TO 241
        IF ( IBLK. EQ. IBSKIP(I,1) ) ILAST = LIMWRD(I,1)
195      IF ( IBLK. EQ. IBSKIP(I,2) ) IFIRST = LIMWRD(I,2)
        IBSWRD = IFIRST
      DO 201 I = IFIRST, ILAST
        IF ( DRDATA(IBSWRD). EQ. NX ) IBSWRD = I
        IF ( DRDATA(I). EQ. NX ) GO TO 201
        IF ( DRDATA(I). LT. DRDATA(IBSWRD) ) IBSWRD = I
201      CONTINUE
C
C *****
C THE CM-1 OF EACH BASELINE PT IS SENT TO FILE "BASLIN.DAT"
C ALONG WITH THE ABSORBANCE AT THIS CM-1.
C
      WVN = ( IBSWRD + WRDDIF - 1 ) / PPWVN
      BASCNT = BASCNT + 1
      WRITE(11,8400) WVN, DRDATA(IBSWRD), M, IBSWRD
      IF ( LASTPT. EQ. 0 ) GO TO 241
      FROMAX = - ( WRDDIF + INDXEP - 1 ) / PPWVN
      WRITE(11,8400) FROMAX
195          GO TO 251
241      CONTINUE
251      CONTINUE
      CLOSE(11)
C *****
C *****
C THE SET OF BASELINE FREQUENCIES AND INTENSITIES PREVIOUSLY
C SENT TO FILE "BASLIN.DAT" ARE READ FROM THIS FILE, AND
C CUBIC SPLINE FITS ARE DONE ON SETS OF 4 CONSECUTIVE BASELINE
C PTS. THE COEFFICIENTS DESCRIBING THE RANGE FROM PTS 2 TO 3
C ARE USED FOR BASELINE INTERPOLATION. THEN, ARRAY INDICES 2-4
C ARE REINDEXED AS 1-3, A SEARCH IS MADE FOR THE NEXT BASELINE
C PT #4, AND ANOTHER CUBIC FIT IS PERFORMED, ETC. FOR THE
C INITIAL AND FINAL SETS OF PTS, ADDITIONAL INTERPOLATIONS ARE
C DONE OUTSIDE OF THE USUAL 2-3 RANGE.
C
261      WRITE(2,6700)
C
C MAKE SURE THAT THE EXPONENT OF THE BASELINE DISPLAY FILE IS
C SET TO A REASONABLE VALUE:
C
      CALL IRTISK( OZDATA, 512, OZFSEC + NSECS, INODE )

```

```

OZDATA(6) = 2
CALL IWTISK( OZDATA, 512, OZFSEC + NSECS, INODE )
C
M      = IISEC
BASCNT = 2
CALL IRTISK( DRDATA, 512, IRFSEC + IISEC, INODE )
READ( 11, 8300 ) BLKSIZ
READ( 11, 8500 ) FRQMIN
INDXSP = NINT( FRQMIN * PPWVN + 1 - (IISEC-1)*512 )
IMIN   = INDXSP
DO 271 I = 1, 4
271   READ( 11, 8400) BFRQ(I), JD, NICSEC(I), NICWRD(I)
      YD(I) = JD
      WRDHI = NINT( BFRQ(1)*PPWVN + 1 - (IISEC-1)*512 )
      WRDLOW = WRDHI
281   IMAX   = 512
      IF ( IMAX. GT. WRDHI) IMAX = WRDHI
C
C FOR THE CM-1 RANGE PRECEDING THE 1ST BASELINE PT, THE
C ABSORBANCE OF THE 1ST BASELINE PT IS SUBTRACTED FROM EACH RAW
C ABSORBANCE VALUE (I.E. CUBIC SPLINES ARE NOT USED).
C
DO 291 I = IMIN, IMAX
      IF ( DRDATA(I). EQ. NX )          GO TO 291
      DRDATA(I) = DRDATA(I) - YD(1)
291   CONTINUE
      IF ( IMAX. EQ. 512 )          GO TO 451
C
C *****
C THE MAIN BASELINE CORRECTION ALGORITHM BEGINS HERE WHENEVER
C THE MOST RECENTLY READ BASELINE PT DOES NOT LIE IN A NEW
C SECTOR OF THE FTIR SCRATCH FILE.
C
301   IMIN = IMAX + 1
311   IF ( IQUIT-1)          321,315,451
C
C CHANGE THE REF PT OF THE FINAL CUBIC EQN FROM PT 3 TO PT 4 FOR
C EXTRAPOLATION OF THE CUBIC FIT BEYOND THE FINAL BASELINE PT.
C
315   D      = BFRQ(4) - BFRQ(3)
      CD(3,1) = 3* CD(3,3) *D*D + 2* CD(3,2) *D + CD(3,1)
      CD(3,2) = 0.0
      IQUIT  = IQUIT + 1
C
321   BASCNT = BASCNT + 1
      IF ( BASCNT. EQ. 4. OR. IQUIT. NE. 0 )   GO TO 421
      IF ( BASCNT. EQ. 3 )                   GO TO 411
C
C FOR EACH BASELINE PT, DRAW A SPIKE IN THE BASELINE DISPLAY
C FILE.
C
      KMIN = 4
      IF ( BASCNT. EQ. 5)   KMIN = 1

```

```

DO 381 K = KMIN, 4
   NSK = NICSEC(K)
   NWK = NICWRD(K)
   IF ( NSK. NE. OLDSEC. AND. OLDSEC. NE. 0 )
1     CALL IWTISK( OZDATA, 512, OZFSEC+OLDSEC, INODE)
   IF ( NSK. NE. OLDSEC )
1     CALL IRTISK( OZDATA, 512, OZFSEC+NSK, INODE )
   OLDSEC = NSK
   OZDATA(NWK) = ZSCALE
DO 351 L = 1, IH
   IF (NWK.GT.L)      OZDATA(NWK-L) = ZSCALE*(1+.1*L)
351  IF (NWK+L.LE.512) OZDATA(NWK+L) = ZSCALE*(1+.1*L)
C
C THE FOLLOWING STEPS DEAL WITH SPIKES IN THE BASELINE DISPLAY
C FILE THAT STRADDLE A SECTOR BOUNDARY:
C
   IF ( NWK+IH. GT. 512. AND. NSK. NE. IFSEC) THEN
     CALL IWTISK( OZDATA, 512, OZFSEC+NSK, INODE )
     CALL IRTISK( OZDATA, 512, OZFSEC+NSK+1, INODE)
     OLDSEC = NSK + 1
     LMAX = NWK - 512 + IH
DO 361  DO 361  L = 1, LMAX
361  OZDATA(L) = ZSCALE* ( 1 + .1*(IH + L - LMAX) )
     ENDIF
   IF ( NWK. LE. IH. AND. NSK. NE. IISEC ) THEN
     CALL IWTISK( OZDATA, 512, OZFSEC + NSK, INODE)
     CALL IRTISK( OZDATA, 512, OZFSEC + NSK-1, INODE)
     OLDSEC = NSK - 1
     LMAX = IH + 1 - NWK
DO 371  DO 371  L = 1, LMAX
371  OZDATA(513-L) = ZSCALE* ( 1 + .1*(IH + L-LMAX))
     ENDIF
381  CONTINUE
     CALL IWTISK( OZDATA, 512, OZFSEC+OLDSEC, INODE)
C
C THE (I+1)TH BASELINE INDEX FOR THE OLD BASELINE ARRAYS
C BECOMES THE ITH BASELINE INDEX FOR THE NEW ARRAYS.
C
DO 401 I = 1, 3
   BFRQ(I) = BFRQ(I+1)
401  YD(I) = YD(I+1)
   READ(11,8400) BFRQ(4), JD, NICSEC(4), NICWRD(4)
   YD(4) = JD
   IF ( BFRQ(4). LT. 0.0 ) THEN
     IQUIT = 1
     BFRQ(4) = -BFRQ(4)
   ENDIF
411  IF ( IQUIT. EQ. 0 ) THEN
     CALL SPLINE( 1, 4, BFRQ, YD, 0.0, 0.0, CD)
   ENDIF
C
C "J" INDICATES WHICH CM-1 VALUE IS THE REFERENCE FOR THE KTH
C SET OF SPLINE COEFFICIENTS. K DIFFERS FROM J ON THE LAST

```


C INTERPOLATION BECAUSE "BFRQ" IS REINDEXED BUT THE "CD" COEFFS
C ARE NOT.

C
421 J = 2 + (IQUIT / 2) - ISTART
 K = 2 + (IQUIT + 1) / 2 - ISTART
 ISTART = 0
 WRDLOW = WRDHI
 WRDHI = NINT(BFRQ(J+1) * PPWVN + 1 - (M-1)*512)

C
C RESUME AT LINE 431 AFTER A NEW SECTOR HAS BEEN READ (I.E WHEN
C INTERPOLATION CORRECTIONS HAVE BEEN INTERRUPTED BY A SECTOR
C BOUNDARY).

C
431 IMAX = 512
 IF (IMAX. GT. WRDHI) IMAX = WRDHI
 DO 441 I = IMIN, IMAX
 IF (DRDATA(I). EQ. NX) GO TO 441
 XD = (I - WRDLOW) / PPWVN
 DBASE=YD(J)+CD(K,1)*XD+CD(K,2)*XD*XD+CD(K,3)*XD**3
 DRDATA(I) = DRDATA(I) - DBASE
441 CONTINUE
 IF (IMAX. NE. 512) GO TO 301
451 CALL IWTISK(DRDATA, 512, ODFSEC + M, INODE)
 M = M + 1
 IF (M. GT. IFSEC) GO TO 1999
 CALL IRTISK(DRDATA, 512, IRFSEC + M, INODE)
 IMIN = 1
 WRDLOW = WRDLOW - 512
 WRDHI = WRDHI - 512
 IF (WRDHI. EQ. 0) GO TO 311

C
C GO TO LINE 281 ONLY IF THE 1ST BASELINE PT OF THE SPECTRUM
C HAS NOT BEEN REACHED YET. SUCH A SITUATION OCCURS ONLY IF THE
C 1ST BASELINE PT DOES NOT LIE IN THE 1ST SECTOR.

C
 IF (ISTART) 1999,431,281
6700 FORMAT('0','BEGIN BASELINE CORRECTIONS')
8000 FORMAT('0','BEGIN BASELINE-LOCATOR ALGORITHM')
8300 FORMAT(I10)
8400 FORMAT(F10.3,3I10)
8500 FORMAT(3F10.3)
1999 CALL EXIT
 END

!
C -----
C "BLANKX.FOR"
C
C CALLS: 1) "FTPARM.FOR"
C 2) FTIR SYSTEM SUBROUTINES
C

```

C CALLING MACRO: BLX
C
C -----
C
C PROGRAM BLANKX
C
C DIMENSION IDATA(512)
C
C READ FTIR PARAMETERS:
C
C NSECS =IRVAL(14001,0)
C INODE =IRVAL(13004,0)
C WVNB =IRVAL(14025,0)
C WVNE =IRVAL(14026,0)
C IDFN =IRVAL(14022,0)
C
C NX =-524288
C IDFSEC=IDFN*NSECS+88-1
C
C READ THE HE-NE CM-1 FROM THE FSB OF FTIR SCRATCH FILE 0, AND
C THEN SET BOUNDARY PARAMETERS:
C
C CALL IRTISK(IDATA,512,ISEC+NSECS,INODE)
C IHENE =IDATA(333)
C CALL FTPARM(IHENE,WVNB,WVNE,FRQMIN,FRQMAX,IISEC,IFSEC,
1 INDXSP,INDXEP,PPWVN)
C
C EXECUTE BLANKING ALGORITHM:
C
C DO 201 M=IISEC,IFSEC
C IWORD=1
C LWORD=512
C IF (M.EQ.IISEC) IWORD=INDXSP
C IF (M.EQ.IFSEC) LWORD=INDXEP
C CALL IRTISK(IDATA,512,IDFSEC+M,INODE)
C DO 101 I=IWORD,LWORD
101 IDATA(I)=NX
201 CALL IWTISK(IDATA,512,IDFSEC+M,INODE)
C CALL EXIT
C END
C
C !
C -----
C "BLANKZ.FOR"
C
C CALLS: FTIR SYSTEM SUBROUTINES
C
C CALLING MACRO: BLZ
C
C -----

```

```

C
  PROGRAM BLANKZ
C
C CALLED BY MACRO "BLZ".  BLANKS SEVERAL SELECTED RANGES OF AN
C ABSORPTION SPECTRUM.  THE BASELINES OF UNBLANKED SEGMENTS ARE
C ADJUSTED TO AVOID SUDDEN OFFSETS IN THE Y-AXIS AT RANGE
C LIMITS.  THE MAIN PURPOSE OF THIS PROGRAM IS TO EXTRACT THE
C BANDS OF A PARTICULAR MOLECULE FROM ALL OTHER BANDS IN A
C SPECTRUM (E.G. THE SPECTRUM OF AN ISOTOPE OF HYDRAZINE CAN BE
C EXTRACTED FROM THE SPECTRUM OF AN ISOTOPIC MIXTURE AND THEN
C PLOTTED FOR A JOURNAL PUBLICATION).
C
C THE INPUT CM-1 RANGES MUST BE INSERTED INTO DATA FILE
C "BLANK.DAT" AS FOLLOWS:
C
C 1) ENTER THE # OF CM-1 RANGES TO BE BLANKED (I4 FORMAT).
C 2) ENTER THE CM-1 RANGES IN ASCENDING ORDER OF CM-1 (2F7).
C 3) REPEAT STEPS 1 & 2 FOR ADDITIONAL SPECTRA.
C
  DIMENSION IDATA(512)
  REAL LXF,NXTFRQ
  OPEN (12,FILE='BLANK.DAT',FORM='FORMATTED',STATUS='OLD')
  DATA R /0.016204542/, ZERO /262144/
C
C READ FTIR PARAMETERS:
C
  IDFN= IRVAL(14022,0)
  NSECS= IRVAL(14001,0)
  INODE= IRVAL(13004,0)
  NTP256= IRVAL(14002,0)
  ID = IRVAL(13764,0)
C
  IDFSEC=IDFN*NSECS+88-1
  PPWVN=R*NTP256
  IF (ID.EQ.1) GO TO 18
C
C READ PAST ALL INPUT DATA PRECEDING THE DESIRED DATA:
C
  DO 15 I=1,ID-1
  READ(12,1001) NRANGE
  DO 12 J=1,NRANGE
12  READ(12,2001) TRASH1,TRASH2
15  CONTINUE
C
18  READ(12,1001) NRANGE
  READ(12,2001) FRQ0,FXF
C
C *****
C BEGIN CM-1 RANGE LOOP:
C
  DO 500 K=1,NRANGE
  IF (K.NE.NRANGE) READ(12,2001) LXF,NXTFRQ
  ICOUNT=0

```

```

C
C COMPUTE WORD INDICES FOR THE BEGINNING OF THE BLANKED RANGE
C (NBEGIN), THE BEGINNING OF THE UNBLANKED RANGE (NPKI), AND
C THE END OF THE UNBLANKED RANGE (NPKF):
C
      NPKI=1+PPWVN*FXF
      NPKF=1+PPWVN*LXF
      NBEGIN=1+PPWVN*FRQO
C
C COMPUTE SECTOR AND ARRAY INDICES CORRESPONDING TO ABOVE WORDS
C
      IISEC=NBEGIN/512+1
      IPISEC=NPKI/512+1
      IFSEC=NPKF/512+1
      LPKLOC=NPKF-512*(IFSEC-1)
      IPKLOC=NPKI-512*(IPISEC-1)
      LOCBEG=NBEGIN-512*(IISEC-1)
C
      IF (K.EQ.NRANGE) IFSEC=IPISEC
      IF (K.EQ.NRANGE) GO TO 27
C
C COMPUTE THE SLOPE AND INTERCEPT OF THE BASELINE CORRECTION:
C
      CALL IRTISK(IDATA,512,IDFSEC+IFSEC,INODE)
      LBASE=IDATA(LPKLOC)-ZERO
      IF (IFSEC.NE.IPISEC) CALL IRTISK(IDATA,512,IDFSEC+IPISEC,
1      INODE)
      IBASE=IDATA(IPKLOC)-ZERO
      ZB=IDATA(IPKLOC)
      SLOPE=(LBASE-IBASE)/(1.0*(NPKF-NPKI))
C
C *****
C PROCESS SECTORS CORRESPONDING TO THE CURRENT CM-1 RANGE:
C
27   DO 100 I=IISEC,IFSEC
      IF (I.NE.IISEC.OR.IISEC.NE.IPISEC.OR.K.EQ.NRANGE)
1     CALL IRTISK (IDATA,512,IDFSEC+I,INODE)
      JI=1
      JL=512
      IF (I.EQ.IISEC)  JI=LOCBEG
      IF (I.EQ.IPISEC) JL=IPKLOC
C
C CASE 1: (I < IPISEC); SECTOR PRECEDES UNBLANKED RANGE; DO
C          BLANKING ONLY.
C CASE 2: (I = IPISEC); SECTOR CONTAINS THE START OF THE
C          UNBLANKED RANGE; DO BLANKING AND THEN
C          BASELINE CORRECTIONS.
C CASE 3: (I > IPISEC); SECTOR BEGINS BEYOND THE START OF THE
C          UNBLANKED RANGE; DO ONLY BASELINE
C          CORRECTIONS.
C
      IF (I.GT.IPISEC) GO TO 55
29   DO 30 J=JI,JL

```

```

30  IDATA(J)=ZERO
    IF (I.NE.IPISEC.OR.K.EQ.NRANGE) GO TO 70
    JI=IPKLOC+1
55  IF (I.GT.IPISEC) JI=1
    JL=LPKLOC
    IF (I.NE.IFSEC) JL=512
    IF (JI.GT.JL) GO TO 70
    DO 60 J=JI,JL
    ICOUNT=ICOUNT+1
60  IDATA(J)=IDATA(J)*ZERO/(ZB+SLOPE*ICOUNT)
70  CALL IWTISK(IDATA,512,IDFSEC+I,INODE)
100 CONTINUE
C *****
C
    FRQO=LXF
    FXF=NXTFRQ
500 CONTINUE
1001 FORMAT(I4)
2001 FORMAT(2F7.2)
    CALL EXIT
    END

```

```

!
C -----
C "EMPFIL.FOR"
C
C CALLS:   FTIR SYSTEM SUBROUTINES
C
C CALLING MACRO: EMP
C -----
C
C   PROGRAM EMPFIL
C
C "EMPFIL.FOR" IS CALLED BY MACRO "EMP".  IT CHECKS THE FSB OF
C FTIR FILE "DFN" TO SEE WHETHER THE FILE IS VACANT OR NOT, &
C SETS FTIR PARAMETER "CMP" TO 0 (FILLED) OR 99 (VACANT).
C
C   INTEGER IW(512)
C
C   IDFN  = IRVAL( 14022, 0 )
C   NSECS = IRVAL( 14001, 0 )
C   INODE = IRVAL( 13004, 0 )
C
C   CALL IRTISK( IW, 512, NSECS * IDFN + 88 + 87, INODE )
C   ICMP = 0
C   IF ( IW(2). NE. 342225 ) THEN
C     ICMP = 99
C     PRINT *, 'FILE #', IDFN, 'WAS VACANT'

```

```

        ENDIF
CALL IRPUT( 13761, 0, ICMP )
CALL EXIT
END

```

```

!
C -----
C "FSBTIT.FOR"
C
C SUBROUTINE OF "PLOTFT.FOR"
C -----
C
C SUBROUTINE FSBTIT( IFILE,TITYPE,NSECS,INODE,IDATA,TITLE,
1
C LETITL )
C
C READS THE TITLE (TITYPE=0) OR MACRO DESCRIPTOR (TITYPE=1) FROM
C THE FILE STATUS BLOCK OF FTIR SCRATCH FILE # "IFILE". THE
C FSB IS LOCATED USING THE INPUT VALUES FOR # OF SECTORS PER
C FILE (NSECS) AS WELL AS THE FTIR "INODE" VALUE. "IDATA" IS
C AN ARRAY USED FOR TEMPORARY DATA STORAGE.
C
C THE OUTPUT IS THE TITLE AND THE # OF LETTERS IN THE TITLE.
C
C INTEGER TITLE(88),IDATA(512),TITYPE
C
C ISEC=(IFILE+1)*NSECS+88-1
C CALL IRTISK(IDATA,512,ISEC,INODE)
C NBLANK=0
C LETITL=0
C
C IF (TITYPE.EQ.0) THEN
C JMIN=68
C JMAX=107
C ELSE
C JMIN=145
C JMAX=232
C ENDIF
C
C CHARACTER CODES FOR FSB CHARACTERS:
C
C 0 OR 1 : NULL
C 141 : 'RETURN' CHARACTER IN FTIR FILE TITLES
C 160 : BLANK
C 166 : &
C 176-185: NUMERALS 0-9
C 193-218: A TO Z
C
C THE TITLE IS TERMINATED WHEN 0, 1, 141, 166, OR 3 SUCCESSIVE
C 160'S ARE ENCOUNTERED (SEE ABOVE CODE TABLE).

```

```

C
DO 101 J=JMIN,JMAX
IF (IDATA(J).LE.0.OR.IDATA(J).EQ.166.OR.
1 IDATA(J).EQ.141) GO TO 201
TITLE(J+1-JMIN)=IDATA(J)
LETITL=LETITL+1
IF (IDATA(J).EQ.160) NBLANK=NBLANK+1
IF (IDATA(J).NE.160) NBLANK=0
IF (NBLANK.GT.2) GO TO 201
101 CONTINUE
201 LETITL=LETITL-NBLANK
RETURN
END

```

```

!
C -----
C "FSPRN.FOR"
C
C CALLS: FTIR SYSTEM SUBROUTINES
C
C CALLING MACRO: FSP
C -----
C
C PRINTS A SECTOR OF FTIR SCRATCH FILE DATA.
C

```

```

PROGRAM FSPRN
INTEGER IW(512)
IDFN =IRVAL(14022,0)
NSECS =IRVAL(14001,0)
INODE =IRVAL(13004,0)
ISEC =IRVAL(14177,0)
CALL IRTISK(IW,512,NSECS*IDFN+ISEC+87,INODE)
IEND=-4
DO 201 J=1,4
WRITE(2,1000) IDFN,ISEC
WRITE(2,3000)
ISTART=IEND+5
IEND=IEND+150
IF (J.EQ.4) IEND=506
DO 101 I=ISTART,IEND,5
101 WRITE(2,2000) I,IW(I),IW(I+1),IW(I+2),
1 IW(I+3),IW(I+4)
IF (J.EQ.4) THEN
WRITE(2,2000) 511,IW(511),IW(512)
ELSE
PRINT *,'TYPE <CR> TO CONTINUE:'
READ *,I
ENDIF

```

```

201          CONTINUE
1000  FORMAT(' ', 'FILE = ', I3, '          SECTOR = ', I3)
2000  FORMAT(' ', I8, 10X, 5I8)
3000  FORMAT('0', ' ')
      CALL EXIT
      END

```

```

!
C -----
C "FSWR.FOR"
C
C CALLS:  FTIR SYSTEM SUBROUTINES
C
C CALLING MACRO: FSW
C
C -----
C
C THIS PROGRAM CHANGES A SINGLE WORD IN THE STATUS BLOCK OF AN
C FTIR SCRATCH FILE.
C

```

```

      PROGRAM FSWR
      INTEGER IW(512), WORD
            INODE = IRVAL(13004, 0)
            IWORD = IRVAL(13762, 0)
            WORD = IRVAL(13763, 0)
            NSECS = IRVAL(14001, 0)
            IDFN = IRVAL(14022, 0)
            CALL IRTISK( IW, 512, NSECS*(IDFN+1)+87, INODE )
            IW(IWORD) = WORD
            CALL IWTISK( IW, 512, NSECS*(IDFN+1)+87, INODE )
      CALL EXIT
      END

```

```

!
C -----
C "FTPARM.FOR"
C
C SUBROUTINE OF MANY FTIR PROGRAMS
C
C -----
C
      SUBROUTINE FTPARM(IHENE, RFXF, RLXF, FRQMIN, FRQMAX, IISEC,
1          IFSEC, INDXSP, INDXEP, PPWVN)
C *****
C
C PURPOSE OF SUBROUTINE:

```



```

C      COMPUTE SPECTRAL RANGE PARAMETERS.
C
C INPUT:
C
C IHENE      : HENE LASER CM-1 MULTIPLIED BY 32 (I.E. THE
C             WORD STORED IN DECIMAL LOCATION 332 OF FSB).
C RFXF,RLXF  : INITIAL AND FINAL WAVENUMBERS OF THE SPECTRAL
C             REGION UNDER CONSIDERATION.
C
C OUTPUT:
C
C FRQMIN,FRQMAX: RFXF AND RLXF ARRANGED IN ASCENDING ORDER.
C IISEC,IFSEC  : RELATIVE INITIAL & FINAL SECTORS CORRESPONDING
C             TO FRQMIN & FRQMAX (WITH SECTOR 1 DEFINED AS
C             BEGINNING AT 0 CM-1 OF THE CURRENT SPECTRUM).
C INDXSP,INDXEP: RELATIVE INITIAL AND FINAL WORDS CORRESPONDING
C             TO FRQMIN AND FRQMAX (EACH REFERENCED TO THE
C             START OF THE SECTOR IN WHICH IT IS LOCATED).
C PPWVN       : RATIO OF WORD RANGE TO WAVENUMBER RANGE.
C *****C
C
C      NTP256= IRVAL(14002,0)
C      LASWVN= IHENE/32
C      R=0.016204542
C      IF (LASWVN.EQ.15798) R=0.016204008
C      PPWVN= R * NTP256
C      FRQMIN=RFXF
C      FRQMAX=RLXF
C      IF (RLXF.LT.RFXF) FRQMIN=RLXF
C      IF (RLXF.LT.RFXF) FRQMAX=RFXF
C
C IFWORD IS OBTAINED BY TRUNCATION.  IIWORD IS OBTAINED BY
C TRUNCATION FOLLOWED BY ADDING AN EXTRA 1.
C
C      IIWORD=2+INT(PPWVN*FRQMIN)
C      IISEC =(IIWORD-1)/512+1
C      INDXSP=IIWORD+(1-IISEC)*512
C      IFWORD=1+INT(PPWVN*FRQMAX)
C      IFSEC =(IFWORD-1)/512+1
C      INDXEP=IFWORD+(1-IFSEC)*512
C      RETURN
C      END
C
C -----
C "INTGRL.FOR"
C
C CALLS:  FTIR SYSTEM SUBROUTINES
C
C CALLING MACRO: ITG

```

```

C -----
C
C PROGRAM INTGRL
C
C "INTGRL.FOR" IS USED IN CONJUNCTION WITH MACRO "ITG" TO
C INTEGRATE SELECTED CM-1 RANGES OF AN ABSORBANCE FILE. THE
C INPUT CM-1 RANGES MUST BE INSERTED INTO DATA FILE
C "INTSEG.DAT" AS FOLLOWS:
C
C 1) ENTER THE # OF CM-1 RANGES TO BE INTEGRATED (I4 FORMAT).
C
C 2) ENTER A BAND ID # (E.G. NORMAL MODE #) & A PAIR OF CM-1
C RANGES ON EACH NEW LINE (I5,2F7 FORMAT).
C
C 3) REPEAT STEPS 1 & 2 FOR ADDITIONAL SPECTRA.
C
C *****
C "INTGRL.FOR" IS CALLED BY MACRO "ITG" & IS WITHIN A MACRO
C DO LOOP. EXCEPT FOR THE FIRST, SECOND, AND LAST ITERATIONS
C OF THIS PROGRAM, EACH ITERATION CONSISTS OF:
C
C 1) READING FROM THE DATA FILE (UNIT 12) THE CM-1 RANGE
C THAT HAS JUST BEEN INTEGRATED IN THE MACRO ROUTINE,
C AND THEN PRINTING THE INTEGRATED INTENSITY FOR THIS
C RANGE.
C
C 2) PUTTING THE NEXT CM-1 RANGE INTO THE PARAMETER
C STORAGE AREA WHERE IT CAN BE ACCESSED BY THE MACRO.
C
C FOR ITERATIONS 1 AND 2, THE FIRST STEP IS SKIPPED. ITERATION
C 1 DEALS WITH THE REFERENCE PEAK, AND 2 DEALS WITH THE 1ST
C CM-1 RANGE. FOR THE LAST ITERATION, STEP 2 IS SKIPPED.
C
C *****
C *****
C INTEGER QIT
C REAL ITG,LXF
C DIMENSION RFXF(50),RLXF(50),NVMODE(50)
C OPEN (12,FILE='INTSEG.DAT',FORM='FORMATTED',STATUS='OLD')
C
C *****
C RETRIEVE THE MACRO PARAMETERS SPECIFYING THE SET OF CM-1
C RANGES TO BE USED (ID), AND THE UPPER MACRO DO LOOP LIMIT
C (QIT). QIT IS 1 UNIT LARGER THAN THE INDEX OF THE FINAL CM-1
C RANGE TO BE PRINTED.
C
C ID = IRVAL(13764,0)
C QIT = IRVAL(14222,0)
C
C OBTAIN THE CURRENT(I=VI1) AND INITIAL(IFIRST=VI3) MACRO CM-1
C LOOP INDICES. "I" CORRESPONDS TO THE CM-1 RANGE THAT WILL
C SUBSEQUENTLY BE INTEGRATED BY THE MACRO. IFIRST IS THE INDEX

```

```

C OF THE FIRST CM-1 RANGE TO BE PRINTED.
C
  I      = IRVAL(13763,0)
  IFIRST = IRVAL(13765,0)
  IROUTE = 0
C
C *****
C READ DATA FILE. THE FIRST ID-1 SETS OF DATA ARE READ BUT
C ARE OVERWRITTEN BY SET ID.
C
  DO 60 K = 1, ID
    READ (12,5000) NLIST
  DO 50 J = 1, NLIST
50    READ(12,4000) NVMODE(J), RFXF(J), RLXF(J)
60    CONTINUE
C
    IF ( I. GT. IFIRST )      GO TO 180
    IF ( I. EQ. IFIRST )      GO TO 90
C *****
C PRINT HEADING.
C
  WRITE (2,1000)
C
  I      = IRVAL(14023,0)
90    IROUTE = 1
                                           GO TO 180
C
C *****
C READ THE INTEGRATED ABSORPTION VALUE WHICH WAS STORED DURING
C THE PREVIOUS ITERATION OF THE MACRO (SKIP THIS STEP ON THE
C FIRST ITERATION OF THIS FORTRAN PROGRAM):
C
100    ITG = RVAL(14062,0)
C
C *****
C SET UP PRINTER OUTPUT:
C
130    REF = RVAL(13766,0)
    REL = ITG / REF
C
    WRITE(2,2000) FXF, LXF, ITG, REL, NMODE
    IF ( I. EQ. QIT) WRITE (2,3000)
    IF ( I. EQ. QIT)      GO TO 200
C
C *****
C SET CM-1 LIMITS FOR PRINTING (IROUTE=0) OR FOR NEXT
C INTEGRATION (IROUTE=1).
C
180    IROUTE = 1
    J      = I - 1 + IROUTE
    FXF    = RFXF(J)
    LXF    = RLXF(J)
    NMODE  = NVMODE(J)

```

```

                IF (IROUTE) 100,100,190
C
C *****
C STORE NEW INTEGRATION LIMITS IN THE PARAMETER STORAGE AREA
C FOR SUBSEQUENT RETRIEVAL BY MACRO ITG.
C
190             FI = FXF
                FF = LXF
                CALL RPUT( 14135, 0, FI )
                CALL RPUT( 14213, 0, FF )
200             CALL EXIT
1000            FORMAT(' ',20X,'FXF',6X,'LXF',8X,'I',7X,'IREL',3X,'MODE #
1')
2000            FORMAT(20X,2(F6.1,3X),2(F6.3,3X),I4)
3000            FORMAT('0',' ')
4000            FORMAT(I5,2F7.1)
5000            FORMAT(I4)
                END

```

```

!
C -----
C "LETTRS.FOR"
C
C CALLS: 1) "GPLOT.LIB" SYSTEM PLOTTING SUBROUTINES
C          2) FTIR SYSTEM SUBROUTINES
C
C CALLING MACRO: LTR
C -----
C
C          PROGRAM LETTRS
C
C THIS PROGRAM SENDS CHARACTERS FROM THE KEYBOARD TO THE
C PLOTTER. IT IS CALLED BY MACRO 'LTR'.
C
C          CHARACTER ANYKEY*10,STRING(80)*1
C
C          CALL PLOTS(3)
C          NCHTOT=0
C          WRITE(2,1100)
C          WRITE(2,1500)
C          READ(2,*) CHR$IZ
C          WRITE(2,400)
C          WRITE(2,1500)
C          READ(2,*) ANGLE
1 DO 101 J=1,1000
C          WRITE(2,600)
C          WRITE(2,700)
C          WRITE(2,800)

```

```

        WRITE(2,900)
        WRITE(2,1000)
        WRITE(2,500)
        WRITE(2,1500)
        READ(2,200) STRING
        NLETRS=0
DO 11 I=1,80
        IC=ICHAR(STRING(I))
        IF (IC.EQ.33) GO TO 21
11      NLETRS=NLETRS+1
        WRITE(2,1300)
                                           GO TO 1
21      WRITE(2,1400) NLETRS
        IF (NLETRS.EQ.0) GO TO 999
        IF (ICONT.EQ.0) THEN
            WRITE(2,1200)
            READ (2,100) ANYKEY
            ENDIF
        IF (ICHAR(STRING(NLETRS+2)).EQ.33) THEN
            ICONT=1
            NCHTOT=NCHTOT+NLETRS
        ELSE
            ICONT=0
        ENDIF
        C=COS(ANGLE*0.017453292)
        S=SIN(ANGLE*0.017453292)
        XSHIFT= CHRSIZ*( (1-ICONT)* 1.3*S + ICONT*NLETRS*C)
        YSHIFT= CHRSIZ*( (1-ICONT)*-1.3*C + ICONT*NLETRS*S)
        CALL SYMBOL(0.0,0.0,CHRSIZ,STRING,ANGLE,NLETRS)
C
C IF THE CURRENT LINE IS FINISHED (I.E. ICONT=0) MOVE TO THE
C NEXT LINE IN THE DIRECTION PERPENDICULAR TO THE CURRENT LINE.
C (NOTE: "CALL PLOT" DOES NOTHING IF ICONT=1.)
C
        CALL PLOT(XSHIFT,YSHIFT,-3)
C
C IF ICONT=0, MOVE BACK TO THE POSITION OF THE 1ST CHARACTER OF
C THE NEXT LINE IN THE DIRECTION PARALLEL TO THE CURRENT
C LINE.
C
        IF (ICONT.EQ.0.AND.NCHTOT.NE.0) THEN
            XSHIFT= -CHRSIZ*NCHTOT*C
            YSHIFT= -CHRSIZ*NCHTOT*S
            NCHTOT=0
            CALL PLOT(XSHIFT,YSHIFT,-3)
            ENDIF
101      WRITE(2,500)
999      CALL PLOT(0,0,999)
100     FORMAT(A10)
200     FORMAT(80A1)
300     FORMAT(F10.0)
400     FORMAT('0','ENTER ANGLE OF LETTERS (0,90,180, OR 270): ')
500     FORMAT('0', '          *          *          *          *')

```

```

/      *          *          *          *')
600  FORMAT('0','TYPE THE CHARACTERS TO BE PLOTTED (MAX OF 80
/ PER LINE);')
700  FORMAT(' ',' END THE STRING WITH EITHER:')
800  FORMAT(' ',' A) AN EXCLAMATION MARK TO EXECUTE [CR] ON
/ THE PLOTTER.')
900  FORMAT(' ',' B) 2 EXCLAMATION MARKS TO CONTINUE THE
/ LINE LATER.')
1000 FORMAT('0','PUT AN EXCLAMATION MARK IN COLUMN #1 TO
/ DISCONTINUE THE PROGRAM:')
1100 FORMAT(' ','ENTER THE CHARACTER SIZE IN INCHES')
1200 FORMAT('0','POSITION THE PLOTTER PEN; THEN TYPE [CR]')
1300 FORMAT('0','ERROR: NO EXCLAMATION MARK FOUND')
1400 FORMAT(' ','# OF LETTERS IN STRING =',I3)
1500 FORMAT(' ','')
      CALL EXIT
      END

```

```

!
C -----
C "MATINV.FOR"
C
C SUBROUTINE OF "SPLINE.FOR"
C -----
C

```

```

      SUBROUTINE MATINV(N,A)
      DIMENSION A(10,10),IP(10,3)
      D=1.0
      DO 51 J=1,N
51      IP(J,3)=0
      DO 901 I=1,N
      AM=0.0
      DO 201 J=1,N
      IF (IP(J,3).EQ.1) GO TO 201
      DO 101 K=1,N
      IF (IP(K,3).EQ.1) GO TO 101
      IF (AM.GE.ABS(A(J,K))) GO TO 101
      IR=J
      IC=K
      AM=ABS(A(J,K))
101      CONTINUE
201      CONTINUE
      IF (AM.LE.1.0E-30) GO TO 1201
      IP(IC,3)=1
      IP(I,1)=IR
      IP(I,2)=IC
      IF (IR.EQ.IC) GO TO 351
      DO 301 L=1,N
      SW=A(IR,L)

```

```

          A(IR,L)=A(IC,L)
301      A(IC,L)=SW
351      PV=A(IC,IC)
          D=D*PV
          A(IC,IC)=1.0
DO 401  L=1,N
401      A(IC,L)=A(IC,L)/PV
DO 601  L1=1,N
          IF (L1.EQ.IC) GO TO 601
          T=A(L1,IC)
          A(L1,IC)=0.0
DO 501  L=1,N
501      A(L1,L)=A(L1,L)-A(IC,L)*T
601      CONTINUE
901      CONTINUE
          NS=0
DO 1101 I=1,N
          L=N-I+1
          IF (IP(L,1).EQ.IP(L,2)) GO TO 1101
          JR=IP(L,1)
          JC=IP(L,2)
          NS=NS+1
DO 1001 K=1,N
          SW=A(K,JR)
          A(K,JR)=A(K,JC)
1001     A(K,JC)=SW
1101     CONTINUE
1201     RETURN
          END

```

```

!
C -----
C "PLOTFT.FOR"
C
C CALLS: 1) "GLOT.LIB" SYSTEM PLOTTING SUBROUTINES
C         2) FTIR SYSTEM SUBROUTINES
C         3) "FSBTIT.FOR"
C         4) "AXIS2.FOR"
C
C CALLING MACROS: PLI & PLC
C
C AUXILIARY PROGRAMS: "PLPARW.FOR" & "PLPARR.FOR"
C -----
C
C PROGRAM PLOTFT
C
C THIS PROGRAM IS CALLED BY MACROS 'PIO' & 'PLC'. IT PRODUCES
C A SINGLE PLOT USING DATA INPUT FROM FTIR PARAMETERS.
C

```

```

C *****
C THE DIMENSIONS OF THE X AND Y ARRAYS ARE 512*SECBLK+2; A MAX
C OF 10 SECTORS ARE PROCESSED EACH TIME SUBROUTINE 'LINE' IS
C CALLED.

```

```

C
C

```

```

      INTEGER      TITLE(88),LTABSC(15),LETORD(15),IDATA(512),
$              PENDAT,PENAX,PENTIT,PLTROT,AUTSCA,
$              ADECDG,WDECDG,ATKRAT,WTKRAT,SECBLK,
$              SECRSM,WRDRSM,RESUME
      REAL         X(5122),Y(5122)
      CHARACTER    ABSCLT(30)*1,ORDLET(30)*1
      DATA        RESUME/0/,SECBLK/10/,
$              ABSCLT
$              /'W','A','V','E','N','U','M','B','E','R','S'
$              ,'N','A','N','O','M','E','T','E','R','S'/,
$              ORDLET
$              /'A','B','S','O','R','B','A','N','C','E'
$              ,'T','R','A','N','S','M','I','T','T','A','N','C','E'/

```

```

C
C *****
C READ INPUT FROM FTIR PARAMETER STORAGE:

```

```

C

```

```

      INODE =IRVAL(13004,0)
      ITER  = RVAL(13702,0)
      ASIZNM=IRVAL(13710,0)*.01
      AUTSCA=IRVAL(13713,0)
      DSTNUM=IRVAL(13714,0)*.01
      WSIZNM=IRVAL(13715,0)*.01
      OFFSET=IRVAL(13731,0)*.01
      SZMJTC=IRVAL(13732,0)*.01
      SIZLET=IRVAL(13733,0)*.01
      SIZTIT=IRVAL(13735,0)*.01
      AANGAX=IRVAL(13736,0)
      AANGNM=IRVAL(13750,0)
      WTKRAT=IRVAL(13751,0)
      ATKRAT=IRVAL(13752,0)
      PLTROT=IRVAL(13753,0)
      PENDAT=IRVAL(13760,0)
      IWAXON=IRVAL(13762,0)
      IAAXON=IRVAL(13763,0)
      ITITON=IRVAL(13764,0)
      PENTIT=IRVAL(13765,0)
      WJSTNM= RVAL(13772,0)
      AJSTNM= RVAL(13774,0)
      VF4    = RVAL(13776,0)
      NSECS =IRVAL(14001,0)
      NTP    =IRVAL(14002,0)*256
      DSTLET=IRVAL(14006,0)*.01
      WDECDG=IRVAL(14012,0)
      WNMPEG= RVAL(14017,0)
      IDFN   =IRVAL(14022,0)
      IOVRLP=IRVAL(14023,0)

```



```

WANGLT=IRVAL(14027,0)
NFILAS=IRVAL(14030,0)
ASTART= RVAL(14031,0)
AEND = RVAL(14033,0)
AANGLT=IRVAL(14035,0)
WANGNM= RVAL(14062,0)
TCSZRT= RVAL(14064,0)
ADECDG=IRVAL(14072,0)
ANMSPC= RVAL(14120,0)
WSTART= RVAL(14135,0)
WTITCO= RVAL(14153,0)
ANMBEG= RVAL(14161,0)
WAXSPN= RVAL(14172,0)
WNMSPC= RVAL(14174,0)
PENAX =IRVAL(14177,0)
APNOO = RVAL(14205,0)
AAXSPN= RVAL(14207,0)
ATITCO= RVAL(14211,0)
WEND = RVAL(14213,0)
WPNO = RVAL(14215,0)
WANGAX=IRVAL(14221,0)
IQIT =IRVAL(14222,0)

```

```

C
C CALCULATE THE Y SPAN (INCHES) OF THE CURRENT PLOT:
C

```

```

IF (IOVRLP.EQ.1) AISPAN = AAXSPN + (1 - IQIT) * VF4
IF (IOVRLP.EQ.0) AISPAN = (AAXSPN + (1 - IQIT)*VF4) / IQIT
IF (VF4.LT.-10) AISPAN = AAXSPN

```

```

C
C CALCULATE THE PEN POSITION OF THE CURRENT PLOT ORIGIN:
C

```

```

IF ( IOVRLP. EQ. 1 ) APNO = ITER * VF4 + APNOO
IF ( IOVRLP. EQ. 0 ) APNO = ITER * ( AISPAN + VF4 ) + APNOO
IF (PLTROT.EQ.0) THEN
    XPNO = WPNO
    YPNO = APNO
ELSE
    XPNO = APNO
    YPNO = WPNO
ENDIF

```

```

C
IF (WEND.GT.WSTART.AND.WNMSPC.LT.0) WNMSPC= -WNMSPC
IF (WEND.LT.WSTART.AND.WNMSPC.GT.0) WNMSPC= -WNMSPC
IF (AEND.GT.ASTART.AND.ANMSPC.LT.0) ANMSPC= -ANMSPC
IF (AEND.LT.ASTART.AND.ANMSPC.GT.0) ANMSPC= -ANMSPC

```

```

C *****
C

```

```

C 'OFFSET' IS THE MINIMUM ALLOWED DISTANCE BETWEEN THE BASELINE
C AND THE CM-1 AXIS IN INCHES. IT IS USED TO KEEP THE BASELINE
C AND THE CM-1 AXIS FROM COINCIDING. 'RNGMLT' IS THE RANGE
C EXPANSION FACTOR COMPUTED FROM 'OFFSET'. THE EXPANDED RANGE
C IS SUBTRACTED FROM THE MAX ABSORBANCE TO OBTAIN A REVISED
C LOWER ABSORBANCE AXIS LIMIT 'ASTART'.

```

```

C
  IF (AUTSCA.EQ.1) THEN
    RNGMLT = AISPAN/(AISPAN-OFFSET)
    ASTART = AEND - RNGMLT*(AEND-ASTART)
    ANMBEG = ASTART
    ANMSPC = (AEND-ASTART)/AISPAN
  ENDIF

C
C INITIALIZE THE PLOTTER & READ THE FILE STATUS BLOCK:
C
  CALL PLOTS(3)
  CALL FSBTIT(IDFN,IT,NSECS,INODE,IDATA,TITLE,LETITL)

C
  NX=-524288
  IF (PENAX.NE.1) CALL NEWPEN(PENAX)
  IPEN=PENAX

C
C *****
C DRAW ABSORBANCE AXIS:
C
  IF (IAAXON.EQ.0)                                GO TO 101
    IT = ITITON - 1
    IF (IDATA(9).EQ.0) THEN
      ILOW = 11
      NLORD = 13
    ELSEIF (IDATA(10).EQ.0) THEN
      ILOW = 1
      NLORD = 10
    ELSE
      NLORD = 0
    ENDIF
    J = 0
    IF (NLORD.EQ.0)                                GO TO 93
  DO 91 I = ILOW, ILOW + NLORD - 1
    J = J + 1
    LETORD(J) = ICHAR(ORDLET(I))
91
C
93  IF (ADECDG.GE.10) ADECDG = 2 - ALOG10(AEND)
C
C FOR COMPARISON PLOTS, THE Y-AXIS IS DRAWN ONLY ONCE.
C THUS, IN THIS CASE, THE Y AXIS IS DRAWN TO THE FULL LENGTH
C "AAXSPN" RATHER THAN THE INDIVIDUAL PLOT LENGTH "AISPAN".
C
  AXSPAN = AISPAN
  IF (AUTSCA.EQ.1.AND.IOVRLP.EQ.1) AXSPAN = AAXSPN

C
  CALL AXIS2 (ASTART, AEND, XPNO, YPNO, AXSPAN,
1          AANGAX, SZMJTC, ATKRAT, TCSZRT, ANMBEG,
2          ANMSPC, ADECDG, ASIZNM, DSTNUM, AJSTNM,
3          AANGNM, LETORD, NLORD, SIZLET, DSTLET,
4          0.0 ,AANGLT)

C
C *****

```

```

C DRAW WAVENUMBER AXIS:
C
101 IF (IWAXON.EQ.0) GO TO 201
      IF (IDATA(11).EQ.0) THEN
            ILOW = 12
            NLABSC = 10
      ELSE
            ILOW = 1
            NLABSC = 11
      ENDIF
      J = 0
      DO 111 I = ILOW, ILOW + NLABSC - 1
            J = J + 1
111     LTABSC(J) = ICHAR(ABSCLT(I))
      CALL AXIS2 (WSTART, WEND, XPNO, YPNO, WAXSPN,
1         WANGAX, SZMJTC, WTKRAT, TCSZRT, WNMPEG,
2         WNMSPC, WDECDG, WSIZNM, DSTNUM, WJSTNM,
3         WANGNM, LTABSC, NLABSC, SIZLET, DSTLET,
4         0.0 ,WANGLT)
C
C *****
C PRINT FILE TITLE ON PLOT IF SUCH PRINTING WAS REQUESTED.
C
201     IF (ITITON.EQ.0) GO TO 235
      IF (PENTIT.NE.IPEN) CALL NEWPEN(PENTIT)
      IPEN = PENTIT
C
C DETERMINE THE PEN LOCATION FOR THE 1ST CHARACTER OF THE
C TITLE. REDUCE THE CURRENT CHARACTER SIZE IF THE TITLE WILL
C OTHERWISE EXCEED THE X-AXIS SPAN:
C
      IF (VF4.LT.SIZTIT.AND.IOVRLP.EQ.1)
1         ATITCO=ATITCO + (0.1+SIZTIT)*ITER
      ATITLC = APNO + AISPAN + ATITCO
      SIZ = SIZTIT
      WRANGE = ABS(WAXSPN)
      IF (LETITL*SIZTIT.GT.WRANGE) SIZTIT = WRANGE/LETITL
C
C CENTERED TITLE:
C
      WTITLC = WPNO + 0.5*WAXSPN - 0.5*(LETITL-1)*SIZTIT
C
C NON-CENTERED TITLE:
C
      IF (WTITCO.GE.-0.0001) WTITLC = WPNO + WTITCO
C
C MATCH THE X,Y PEN AXES WITH THE X,Y PLOT AXES:
C
      XPNTIT = WTITLC*(1-PLTROT) + ATITLC*PLTROT
      YPNTIT = WTITLC*PLTROT + ATITLC*(1-PLTROT)
C
      CALL SYMBOL(XPNTIT,YPNTIT,-SIZTIT,TITLE,WANGAX,LETITL)
C

```

C *****
C SET PARAMETERS REQUIRED FOR PLOTTING SUBROUTINE 'LINE':

C
235 IF (PENDAT.NE.IPEN) CALL NEWPEN(PENDAT)

C
C COMPUTE VALUES NEEDED FOR LAST 2 ELEMENTS OF X AND Y PLOTTING
C ARRAYS OF SUBROUTINE 'LINE'.
C

```
XSTART = WSTART*(1-PLTROT) + ASTART*PLTROT
YSTART = WSTART*PLTROT      + ASTART*(1-PLTROT)
WPINCH = (WEND-WSTART)/WAXSPN
APINCH = (AEND-ASTART)/AISPN
XPINCH = WPINCH*(1-PLTROT) + APINCH*PLTROT
YPINCH = APINCH*(1-PLTROT) + WPINCH*PLTROT
```

C
C COMPUTE SECTOR LIMITS, WORD LIMITS, AND INTENSITY SCALING
C FACTOR NEEDED FOR READING DATA FROM FTIR SCRATCH FILE:
C

```
SCLFCT = 1/2.0**(19.0-IDATA(6))
IF (IDATA(7).EQ.0) SCLFCT = SCLFCT/IDATA(3)
IF (IDATA(9).EQ.0) SCLFCT = 100*SCLFCT
AWEND = AEND/SCLFCT
IF (IDATA(11).EQ.0) THEN
    IWL = IDATA(24)
    ABSCSP = (NTP/2.0-1)/(IDATA(25)-IWL)
    IWORD = (WSTART-IWL)*ABSCSP + 1
    LWORD = (WEND -IWL)*ABSCSP + 1
ELSE
    BNDWID = 15798/IDATA(17)
    ABSCSP = (NTP/2-1)/BNDWID
    IWORD = WSTART*ABSCSP + 1
    LWORD = WEND *ABSCSP + 1
    IWL = 0
ENDIF
IDIR = 1
IF (IWORD.GT.LWORD) IDIR = -1
IDFSEC = IDFN*NSECS + 88 - 1
ISEC = IWORD/512 + 1
INIWRD = IWORD - 512*(ISEC-1)
LSEC = LWORD/512 + 1
LSTWRD = LWORD - 512*(LSEC-1)
NLCALL = 1 + IDIR*(LSEC-ISEC)/SECBLK
ISI = ISEC
```

C
C SPECTRAL INTENSITIES ARE READ FROM THE DESIGNATED FTIR SCRATCH
C FILE & TRANSFERRED ALONG WITH THEIR ASSOCIATED CM-1'S INTO
C THE X AND Y PLOTTING ARRAYS. IF THE SPECTRAL DATA FILE
C EXCEEDS 'SECBLK' SECTORS, THE DATA ARE PROCESSED IN BLOCKS
C CONTAINING SECBLK SECTORS EACH.
C

```
CALL PLOT(XPNO,YPNO,-3)
DO 291 J = 1,NLCALL
    ISL = ISI + (SECBLK-1)*IDIR
```

```

                IF (J.EQ.NLCALL) ISL = LSEC
261             IP = 0
C
C SPECTRAL INTENSITIES ARE READ FROM THE DESIGNATED FTIR SCRATCH
C FILE & TRANSFERRED ALONG WITH THEIR ASSOCIATED CM-1'S INTO
C THE X AND Y PLOTTING ARRAYS:
C
      DO 281 IS = ISI, ISL, IDIR
        NBLANK = 0
        IF (RESUME.EQ.1) GO TO 266
        CALL IRTISK(IDATA, 512, IDFSEC+IS, INODE)
        IF (IDIR.EQ.1) THEN
          ISTART = 1
          IEND = 512
        ELSE
          ISTART = 512
          IEND = 1
        ENDIF
        IF (IS.EQ.ISI.AND.J.EQ.1) ISTART = INIWRD
        IF (IS.EQ.ISL.AND.J.EQ.NLCALL) IEND = LSTWRD
266       RESUME = 0
      DO 271 I = ISTART, IEND, IDIR
        IP = IP+1
C
C IF BLANKED DATA ARE ENCOUNTERED, SEARCH FOR THE END OF THE
C BLANKED DATA, AND SKIP FURTHER FILLING OF THE SPECTRAL ARRAYS.
C USE "NBLANK" TO COUNT THE # OF BLANKED PTS IN THE CURRENT
C BLANKED REGION; "IBLANK" TO RECORD THE INDEX OF THE 1ST
C BLANKED PT IN THE CURRENT X,Y ARRAYS; AND "IP" TO SPECIFY THE
C # OF PTS IN THE CURRENT BLOCK OF DATA.
C
C Y-AXIS VALUES THAT EXCEED THE SPECIFIED Y-AXIS MAXIMUM ARE
C TREATED AS IF THEY WERE BLANKED PTS.
C
      IF (IDATA(I).EQ.NX.OR.IDATA(I).GT.AWEND) THEN
        IF (NBLANK.EQ.0) IBLANK = IP
        NBLANK = NBLANK+1
C
C IF GOOD DATA ARE FOUND AFTER A BLANKED REGION HAS BEEN
C ENCOUNTERED, SAVE THE SECTOR AND WORD INDICES OF THE START OF
C THE GOOD DATA, AND THEN PLOT THE REGION (IF ANY) THAT
C PRECEDED THE BLANKED REGION.
C
      ELSEIF (NBLANK.NE.0) THEN
        SECRSM = IS
        WRDRSM = I
        IP = IBLANK-1
        IF (IP.GT.0) GO TO 285
        IF (IP.LE.0) GO TO 287
      ELSE
        ABSCVL = (512*(IS-1)+I-1)/ABSCSP + IWL
        X(IP) = SCLFCT*IDATA(I)*PLTROT
        + ABSCVL*(1-PLTROT)

```

\$

```

                Y(IP) = SCLFCT*IDATA(I)*(1-PLTROT)
                + ABSCVL*PLTROT
$
                ENDIF
271          CONTINUE
C
C IF A BLANKED REGION EXTENDS TO THE END OF A BLOCK OF PLOTTING
C DATA, RESET THE BLOCK INDEX TO THE LAST GOOD DATA POINT AND
C THEN PLOT THE DATA.
C
                IF (NBLANK.NE.0) THEN
                    IP = IBLANK-1
                    NBLANK = 0
                ENDIF
281          CONTINUE
C
C SET THE PEN ORIGIN EQUAL TO THE AXIS ORIGIN & THEN PLOT A
C BLOCK OF DATA. SINCE THE LAST 2 ELEMENTS OF THE X & Y ARRAYS
C ARE ALWAYS DETERMINED FROM THE OVERALL CM-1 BOUNDARIES RATHER
C THAN THE BLOCK BOUNDARIES, THE PEN ORIGIN IS NOT SHIFTED FOR
C EACH NEW BLOCK.
C
285          X(IP+1) = XSTART
                Y(IP+1) = YSTART
                X(IP+2) = XPINCH
                Y(IP+2) = YPINCH
                IF (IP.GT.1) CALL LINE(X,Y,IP,1,0,2)
C
C IF GOOD DATA FOLLOW BLANKED DATA, RESET INDICES TO WHERE THE
C GOOD DATA BEGIN, AND RESTART FILLING THE SPECTRAL ARRAYS.
C
287          IF (NBLANK.EQ.0)                                GO TO 291
                ISI = SECRSM
                ISTART = WRDRSM
                RESUME = 1
                                                    GO TO 261
291          ISI = ISL + IDIR
C
C MOVE THE PEN BACK TO THE ORIGINAL ORIGIN AND RESET THE ORIGIN
C AT THAT POINT. THEN, CLOSE THE SPECTRAL DATA FILE:
C
                CALL PLOT(-XPNO,-YPNO,999)
                                                    GO TO 1000
700          PRINT *,'FILE READ ERROR IN FILE STATUS BLOCK'
                                                    GO TO 1000
800          PRINT *,'FILE READ ERROR DURING READ SKIP'
                                                    GO TO 1000
900          PRINT *,'FILE READ ERROR IN MAIN DATA READ'
1000         CALL EXIT
                END

```

```

!
C -----
C "PLPARR.FOR"
C
C CALLS: FTIR SYSTEM SUBROUTINES
C
C CALLING MACRO: PLI & PLC
C
C READS INPUT DATA FOR "PLOTFT.FOR"
C -----
C
C          PROGRAM PLPARR
C
C THIS PROGRAM IS CALLED BY CONDITIONAL MACRO "M49" WHICH IS A
C SUB-MACRO OF BOTH "PIO" & "PLC". IT READS FTIR PLOT
C PARAMETERS FROM PRE-EXISTING NICOS FILE "PLT.DAT".
C
C          CHARACTER X*1
C          OPEN(11,FILE='PLT.DAT',FORM='FORMATTED',STATUS='OLD')
C
C          READ(11,100) X
C          READ(11,101) RFXF
C          READ(11,101) RLXF
C          READ(11,100) X
C          READ(11,101) RLXI
C          READ(11,100) X
C          READ(11,101) RCXL
C          READ(11,100) X
C          READ(11,101) RFXL
C          READ(11,101) RLYT
C          READ(11,100) X
C          READ(11,102) ISMN
C          READ(11,100) X
C          READ(11,102) IXSP
C          READ(11,102) IXEP
C          READ(11,100) X
C          READ(11,102) IWTY
C          READ(11,100) X
C          READ(11,102) IMNT
C          READ(11,100) X
C          READ(11,102) IRTR
C          READ(11,100) X
C          READ(11,101) RLXL
C          READ(11,101) RLYA
C          READ(11,100) X
C          READ(11,102) IPEK
C          READ(11,100) X
C          READ(11,102) ISRT
C          READ(11,102) IRTN
C          READ(11,100) X
C          READ(11,101) RFCD
C          READ(11,102) IRTO
C          READ(11,100) X

```

```

READ(11,102) IXSL
READ(11,102) IYSL
READ(11,100) X
READ(11,102) IRTP
READ(11,100) X
READ(11,102) IVI2
READ(11,100) X
READ(11,102) INSS
READ(11,100) X
READ(11,102) IQIT
READ(11,100) X
READ(11,102) IOFN
READ(11,100) X
READ(11,101) RVF4
CALL RPUT(14135,0,RFXF)
CALL RPUT(14213,0,RLXF)
CALL RPUT(14017,0,RLXI)
CALL RPUT(14174,0,RCXL)
CALL RPUT(14172,0,RFXL)
CALL RPUT(14207,0,RLYT)
CALL IRPUT(13713,0,ISMN)
CALL IRPUT(14025,0,IXSP)
CALL IRPUT(14026,0,IXEP)
CALL IRPUT(13710,0,IWTY)
CALL IRPUT(13731,0,IMNT)
CALL IRPUT(13753,0,IRTR)
CALL RPUT(14215,0,RLXL)
CALL RPUT(14205,0,RLYA)
CALL IRPUT(13760,0,IPEK)
CALL IRPUT(14221,0,ISRT)
CALL IRPUT(13736,0,IRTN)
CALL RPUT(14062,0,RFCD)
CALL IRPUT(13750,0,IRTO)
CALL IRPUT(14027,0,IXSL)
CALL IRPUT(14035,0,IYSL)
CALL IRPUT(13751,0,IRTP)
CALL IRPUT(13764,0,IVI2)
CALL IRPUT(14004,0,INSS)
CALL IRPUT(14222,0,IQIT)
CALL IRPUT(14023,0,IOFN)
CALL RPUT(13776,0,RVF4)
100  FORMAT(64A1)
101  FORMAT(5X,F7.1)
102  FORMAT(5X,I5)
999  CALL EXIT
      END

```

!


```

C -----
C "PLPARW.FOR"
C
C CALLS: FTIR SYSTEM SUBROUTINES
C
C CALLING MACROS: PLI & PLC
C
C CREATES AN INPUT DATA FILE FOR "PLOTFT.FOR"
C -----
C
C PROGRAM PLPARW
C
C THIS PROGRAM IS CALLED BY CONDITIONAL MACRO "M48" WHICH, IN
C TURN, IS CALLED BY MACROS "PLI" & "PLC". IT SAVES FTIR PLOT
C PARAMETERS IN A NICOS FILE FOR LATER USE.
C
C INTEGER*2 ISIZE
C OPEN(11,FILE='PLT.DAT',FORM='FORMATTED',STATUS='UNKNOWN',
C $SIZE=ISIZE)
C
C RFXF = RVAL(14135,0)
C RLXF = RVAL(14213,0)
C RLXI = RVAL(14017,0)
C RCXL = RVAL(14174,0)
C RFXL = RVAL(14172,0)
C RLYT = RVAL(14207,0)
C ISMN =IRVAL(13713,0)
C IXSP =IRVAL(14025,0)
C IXEP =IRVAL(14026,0)
C IWTY =IRVAL(13710,0)
C IMNT =IRVAL(13731,0)
C IRTR =IRVAL(13753,0)
C RLXL = RVAL(14215,0)
C RLYA = RVAL(14205,0)
C IPEK =IRVAL(13760,0)
C ISRT =IRVAL(14221,0)
C IRTN =IRVAL(13736,0)
C RFGD = RVAL(14062,0)
C IRTO =IRVAL(13750,0)
C IXSL =IRVAL(14027,0)
C IYSL =IRVAL(14035,0)
C IRTP =IRVAL(13751,0)
C IVI2 =IRVAL(13764,0)
C INSS =IRVAL(14004,0)
C IQIT =IRVAL(14222,0)
C IOFN =IRVAL(14023,0)
C RVF4 = RVAL(13776,0)
C WRITE(11,100)
C WRITE(11,101) RFXF
C WRITE(11,102) RLXF
C WRITE(11,103)
C WRITE(11,104) RLXI
C WRITE(11,105)

```

```

WRITE(11,106) RCXL
WRITE(11,107)
WRITE(11,108) RFXL
WRITE(11,109) RLYT
WRITE(11,110)
WRITE(11,111) ISMN
WRITE(11,112)
WRITE(11,1130) IXSP
WRITE(11,1140) IXEP
WRITE(11,1141)
WRITE(11,1142) IWTY
WRITE(11,115)
WRITE(11,1160) IMNT
WRITE(11,1170)
WRITE(11,1180) IRTR
WRITE(11,119)
WRITE(11,1200) RLXL
WRITE(11,1210) RLYA
WRITE(11,1240)
WRITE(11,1250) IPEK
WRITE(11,126)
WRITE(11,1270) ISRT
WRITE(11,1280) IRTN
WRITE(11,129)
WRITE(11,1310) RFCD
WRITE(11,1320) IRTO
WRITE(11,1330)
WRITE(11,1340) IXSL
WRITE(11,1350) IYSL
WRITE(11,1360)
WRITE(11,1370) IRTP
WRITE(11,1380)
WRITE(11,1390) IVI2
WRITE(11,1400)
WRITE(11,1410) INSS
WRITE(11,1420)
WRITE(11,1430) IQIT
WRITE(11,1440)
WRITE(11,1450) IOFN
WRITE(11,1480)
WRITE(11,1490) RVF4
100 FORMAT('WAVENUMBERS OF 1ST & LAST DATA PTS:')
101 FORMAT('FXF ',F7.1)
102 FORMAT('LXF ',F7.1)
103 FORMAT('1ST TIC LABEL ON CM-1 AXIS:')
104 FORMAT('LXI ',F7.1)
105 FORMAT('CM-1 SPACING OF TIC LABELS:')
106 FORMAT('CXL ',F7.1)
107 FORMAT('AXES LENGTHS FOR CM-1 (FXL) & ABSORBANCE (LYT):')
108 FORMAT('FXL ',F7.1)
109 FORMAT('LYT ',F7.1)
110 FORMAT('AUTOSCALING: 0=NONE, 1=ALL, >1 = FILE SMN SCALE')
111 FORMAT('SMN ',I5)

```

```

112  FORMAT('CM-1 LIMITS FOR AUTOSCALE:')
1130  FORMAT('XSP  ',I5)
1140  FORMAT('XEP  ',I5)
1141  FORMAT('DIGIT SIZE FOR ABSORBANCE LABELS; - KILLS LABELS')
1142  FORMAT('WTY  ',I5)
115  FORMAT('MARGIN (.01 INCHES) BETWEEN BASELINE & CM-1 AXIS')
1160  FORMAT('MNT  ',I5)
1170  FORMAT('ROTATE PLOT (0=NO, 1=YS)?')
1180  FORMAT('RTR  ',I5)
119  FORMAT('ORIGIN (INCHES) OF CM-1 (LXL) & ABSORBANCE (LYA)
* AXES VS. ZPN:')
1200  FORMAT('LXL  ',F7.1)
1210  FORMAT('LYA  ',F7.1)
1240  FORMAT('INITIAL PEN # FOR SPECTRAL DATA:')
1250  FORMAT('PEK  ',I5)
126  FORMAT('ANGLE OF AXIS FOR CM-1 (SRT) & ABSORBANCE (RTN):')
1270  FORMAT('SRT  ',I5)
1280  FORMAT('RTN  ',I5)
129  FORMAT('ANGLE OF AXIS LABELS FOR CM-1 (FCD) & ABS (RTO):')
1310  FORMAT('FCD  ',F7.1)
1320  FORMAT('RTO  ',I5)
1330  FORMAT('ANGLE OF AXIS NAME FOR CM-1 (XSL) & ABS (YSL):')
1340  FORMAT('XSL  ',I5)
1350  FORMAT('YSL  ',I5)
1360  FORMAT('RATIO OF MINOR TO MAJOR TICS:')
1370  FORMAT('RTP  ',I5)
1380  FORMAT('PLOT TITLES (0=NO, 1=YES)?')
1390  FORMAT('VI2  ',I5)
1400  FORMAT('SPECTRUM PEN # INCREMENT FOR SUCCESSIVE PLOTS:')
1410  FORMAT('NSS  ',I5)
1420  FORMAT('# OF PLOTS:')
1430  FORMAT('QIT  ',I5)
1440  FORMAT('OVERLAP Y-AXES (0=NO, 1=YES) ?')
1450  FORMAT('OFN  ',I5)
1480  FORMAT('Y MARGIN BETWEEN PLOTS')
1490  FORMAT('VF4  ',F7.1)
CALL EXIT
END

```

```

!
C -----
C "PMAC.FOR"
C
C CALLS: NONE
C -----
C
PROGRAM PMAC
INTEGER LI(200),N(200),PGESIZ
CHARACTER CHR(64)*1,OUT(64)*1,TEXT(60,64)*1,BL*1,IFILE*64
$          ,TRIM*64,IFIL*64,ANYKEY*10

```

DATA BL/' '/

C

C PROMPT FOR KEYBOARD INPUT:

C

```
WRITE(2,1000)
WRITE(2,900)
READ(1,300) IFILE
IFIL = TRIM( IFILE, 64 )
OPEN(11,FILE= IFIL ,FORM='FORMATTED',STATUS='OLD')
WRITE(2,400)
READ(1,*) LPPAGE
WRITE(2,500)
READ(1,*) LSEP
WRITE(2,600)
READ(1,*) IUNPAK
WRITE(2,700)
READ(1,*) IPGSTP
IF ( IPGSTP .NE. 1) IPGSTP = 0
WRITE(2,1200)
WRITE(2,1300)
WRITE(2,1400)
READ(1,*) NBLANK
PGESIZ = NBLANK + LPPAGE
IF (LSEP.LT.1.OR.LSEP.GT.LPPAGE) LSEP = 3
IF (IUNPAK.NE.1) IUNPAK = 0
LFLAG = 0
MARGIN = PGESIZ - LPPAGE
```

C

C READ 1ST LINE OF MACRO LISTING; THEN SKIP TO LINE 161:

C

```
READ(11,100,END=8999) CHR
L = 0
```

GO TO 161

C INSERT "LSEP" BLANK LINES BETWEEN PREVIOUS & CURRENT MACRO:

C

21

```
LFLAG = (L - 1) * IUNPAK
JMAX = LSEP + L
IF (L+LSEP+1.GT.LPPAGE) JMAX = LPPAGE
IF ( L + 1 . GT. JMAX )
```

GO TO 61

```
DO 51 J = L + 1, JMAX
DO 41 I = 1, 64
```

41

```
TEXT(J,I) = BL
```

51

```
CONTINUE
```

C

C IF SUFFICIENT LINES HAVE ACCUMULATED, PRINT A PAGE OF TEXT.

C

61

```
IF (L+LSEP+1.GE.LPPAGE) THEN
DO 121 J = 1, L
```

```
DO 111 I = 1, 64
```

111

```
OUT(I) = TEXT(J,I)
```

```

121             WRITE(10,200) OUT
                CONTINUE
DO 131 J = L + 1, PGESIZ
131             WRITE(10,200) BL
                L = 0
                LFLAG = 0
                IF (IPGSTP.EQ.1) THEN
                    WRITE(2,800)
                    READ(1,1100) ANYKEY
                    ENDIF
                ELSE
                    L = L + LSEP
                ENDIF
C
C SAVE THE CURRENT MACRO HEADER LINE IN THE "TEXT" ARRAY:
C
161             L = L + 1
                DO 171 I=1,64
171             TEXT(L,I) = CHR(I)
C
C CONTINUE READING THE FILE UNTIL EITHER:
C 1) AN "END" STATEMENT IS FOUND,
C 2) "LPPAGE" LINES HAVE BEEN READ.
C
C FOR CASE 1, IGNORE SUBSEQUENT LINES UNTIL AN EXCLAMATION MARK
C IS FOUND.
C
181             READ(11,100,END=8999) CHR
                I1 = ICHAR(CHR(1))
                I2 = ICHAR(CHR(2))
                I3 = ICHAR(CHR(3))
                IF (I1.EQ.69. AND. I2.EQ.78. AND. I3.EQ.68) THEN
                    L = L + 1
                    DO 185 I= 1, 64
185             TEXT(L,I) = CHR(I)
                    IGAP = 1
                    IF ( L. EQ. L ) GO TO 181
                    ENDIF
                    IF (IGAP.EQ.1.AND.I1.NE.33) GO TO 181
                    IGAP = 0
C
C IF ONE EXCLAMATION IS FOUND, GO TO LINE 21.
C IF TWO EXCLAMATIONS ARE " , " " " 201 & PRINT OUTPUT.
C OTHERWISE, ADD THE CURRENT LINE TO THE "TEXT" ARRAY & THEN
C EITHER PRINT A FULL PAGE OR RETURN TO LINE 181.
C
                IF ( I1. EQ. 33. AND. I2. EQ. 33 ) GO TO 201
                IF ( I1. EQ. 33. AND. I2. NE. 33 ) GO TO 21
C
                L = L + 1
                DO 191 I= 1, 64
191             TEXT(L,I) = CHR(I)
C

```

C IF THE "IUNPAK = 1" OPTION HAS BEEN CHOSEN (I.E. LFLAG.NE.0)
C DON'T START PRINTING A MACRO IF IT BEGINS IN THE MIDDLE OF A
C PAGE AND EXTENDS BEYOND THE END OF THAT PAGE. INSTEAD, PRINT
C THE PREVIOUS MACRO(S), RESHUFFLE THE INDICES OF "TEXT", AND
C START A FRESH PAGE.

```
C
201      IF (L.EQ.LPPAGE.AND.LFLAG.NE.0.AND.I2.NE.33) THEN
           JMAX = LFLAG
           ELSE
           JMAX = L
           ENDIF
```

C
C IF "LPPAGE" LINES HAVE BEEN READ OR IF "!!" IS ENCOUNTERED,
C TRANSFER DATA TO THE OUTPUT ARRAY AND PRINT IT.

```
C
           IF ( I2. EQ .33. OR. L. EQ. LPPAGE ) THEN
               DO 221 J = 1, JMAX
               DO 211 I = 1, 64
211          OUT(I) = TEXT(J,I)
               WRITE(10,200) OUT
221          CONTINUE
               DO 231 J = JMAX + 1, PGESIZ
231          WRITE(10,200) BL
                   IF (IPGSTP.EQ.1) THEN
                       WRITE(2,800)
                       READ(1,1100) ANYKEY
                   ENDIF
           ENDIF
```

```
C
           IF (I2.EQ.33) GO TO 9999
```

C
C SKIP THE FOLLOWING "IF" UNLESS IUNPAK=1 (I.E. LFLAG.NE.0),
C AND A PAGE HAS JUST BEEN PRINTED:

```
C
           IF (L.EQ.LPPAGE.AND.LFLAG.NE.0) THEN
               DO 251 J = LFLAG+1, LPPAGE
               DO 241 I = 1, 64
                   JJ = J - LFLAG
                   TEXT(JJ,I) = TEXT(J,I)
241          TEXT(J,I) = BL
251          CONTINUE
                   L = LPPAGE - LFLAG
                   LFLAG = 0
           ENDIF
```

C
C SKIP THE FOLLOWING "IF" UNLESS IUNPAK=0 AND A PAGE HAS JUST
C BEEN PRINTED:

```
C
           IF (L.EQ.LPPAGE.AND.LFLAG.EQ.0) L = 0 GO TO 181
8999 PRINT *, 'ERROR: EOF W/O TWO EXCLAMATION MARKS'
100  FORMAT(64A1)
200  FORMAT(1X,64A1)
```

```

300  FORMAT(A64)
400  FORMAT(1X,'# OF PRINTED LINES PER PAGE: ')
500  FORMAT(1X,'MINIMUM LINE SEPARATION BETWEEN MACROS: ')
600  FORMAT(1X,'START LONG MACROS ON NEW PAGE (0=NO, 1=YS)? ')
700  FORMAT(1X,'PAUSE AFTER EACH PAGE (0=NO, 1=YS)? ')
800  FORMAT(1X,'TYPE [CR] TO CONTINUE')
900  FORMAT('0','INPUT FILE NAME: ')
1000 FORMAT(1X,'*** THIS PROGRAM PRINTS MULTIPLE MACROS FROM
$ A NICOS INPUT FILE ***')
1100 FORMAT(A10)
1200 FORMAT(1X,'# OF BLANK LINES NEEDED TO MATCH THE BLOCK OF
$ PRINTED LINES WITH THE')
1300 FORMAT(1X,'PAGE SIZE OF THE PRINTER (DO NOT COUNT THE 6
$ LINES THAT PRINTERS')
1400 FORMAT(1X,'AUTOMATICALLY SKIP ON EACH SIDE OF PAGE
$ PERFORATIONS: ')
9999  STOP
      END

```

```

!
C -----
C "SBFTYP.FOR"
C
C CALLS:  FTIR SYSTEM SUBROUTINES
C
C CALLING MACROS:  AOB & SBS
C -----
C          PROGRAM SBFTYP
C
C "SBFTYP.FCP" IS CALLED BY MACROS "AOB" & "SBS" TO:
C
C 1) RECLASSIFY FTIR SCRATCH FILE "DFN" FROM %T TO 1-BEAM.
C 2) COMPUTE A MULTIPLIER FOR THE OUTPUT SCRATCH FILE.
C
      INTEGER IFSB(512),VIO
      INODE = IRVAL(13004,0)
      IBOUT = IRVAL(13761,0)
      VIO   = IRVAL(13762,0)
      ABSPT = RVAL(13766,0)
      SBIPT = RVAL(13770,0)
      NSECS = IRVAL(14001,0)
      NSD   = IRVAL(14006,0)
      IDFN  = IRVAL(14022,0)
      CALL IRTISK( IFSB, 512, NSECS * (IDFN + 1) + 87, INODE)
      IFSB(9) = -1
      IFSB(7) = 0
      CALL IWTISK( IFSB, 512, NSECS * (IDFN + 1) + 87, INODE)
      IF ( IBOUT. EQ. 0 )  FCD = SBIPT * 10 ** ( - ABSPT )
      IF ( IBOUT. EQ. 1 )  FCD = SBIPT * 10 ** (  ABSPT )

```



```

NSECS = IRVAL(14001,0)
NTP   = IRVAL(14002,0) * 256
ISFN  = IRVAL(14015,0)
IDFN  = IRVAL(14022,0)
WSMIN = IRVAL(14025,0)
WSMAX = IRVAL(14026,0)
C *****
C
C LOCATE SECTORS TO BE READ USING THE FOLLOWING INDEX:
C
    IDFSEC = IDFN * NSECS + 87
    ISFSEC = ISFN * NSECS + 87
C
C 1) COPY THE FILE STATUS BLOCK OF THE INPUT FILE TO THE OUTPUT
C    FILE.
C 2) IDENTIFY THE FILE AS A CM-1 FILE (FILTYP = 0) OR A
C    WAVELENGTH (FILTYP = 1) FILE.
C
    CALL IRTISK( IDATA, 512, ISFSEC + NSECS, INODE )
C
    IF ( IDATA(11) . EQ. 0 ) THEN
        FILTYP = 1
        WMIN   = IDATA(24)
        WMAX   = IDATA(25)
        PPW    = ( NTP / 2 - 1 ) / ( WMAX - WMIN )
    ELSE
        FILTYP = 0
        WMIN   = 0
        PPW    = 4.1482256 * NTP / 65536
    ENDIF
C
    CALL IWTISK( IDATA, 512, IDFSEC + NSECS, INODE )
C
C COMPUTE THE GAUSSIAN COEFFICIENTS CORRESPONDING TO THE GIVEN
C HALF-WIDTH AT HALF-MAXIMUM.
C
    HWHM = HWHM * PPW
    B    = 0.470 / HWHM
    A    = -0.693 / (HWHM*HWHM)
C
C COMPUTE THE GAUSSIAN MULTIPLIERS; TRUNCATE THE GAUSSIAN AT
C SOME ARBITRARY POINT WHERE THE COEFFICIENTS BECOME
C NEGLIGIBLE. THE CONVOLUTION INTEGRAL IS ALREADY NORMALIZED
C SINCE THE DATA POINT SPACING IS 1 UNIT.
C
    WRITE(2,1000)
1000  FORMAT('0','THE GAUSSIAN COEFFICIENTS ARE:')
    N  = 1
    1  C(N) = B*2.7183**(A*(N-1)*(N-1))
        IF (C(N).LT.0.01*C(1).OR.N.GE.100) GO TO 3
        WRITE(2,2000) N,C(N)
2000  FORMAT('0',I4,F10.3)
    N = N + 1

```

GO TO 1

```
3      NC = N
      NC1 = NC - 1
C
C COMPUTE INITIAL (IISEC) & FINAL (IFSEC) SECTORS:
C
      IF (WSMIN.GT.WSMAX) THEN
          TEMP = WSMIN
          WSMIN = WSMAX
          WSMAX = TEMP
      ENDIF
C
      IIWORD = 1 + PPW * (WSMIN - WMIN)
      IISEC = ( IIWORD - 1 ) / 512 + 1
      IFWORD = 1 + PPW * (WSMAX - WMIN)
      IFSEC = ( IFWORD - 1 ) / 512 + 1
C
C PROCESS THE INITIAL SECTOR SEPARATELY SINCE IT MAY CONTAIN
C BLANKED POINTS.
C
      NSHIFT = 2 * NC1
      ISHIFT = 512 - NSHIFT
      MAXSMO = 512 - NC1
      CALL IRTISK( IDATA, 512, ISFSEC + IISEC, INODE )
      IPT = 1
      DO 100 J=1,512
          IF (IDATA(J).GE.-524287) GO TO 100
          IPT = J + 1
          IDATA(J) = 0
100      JDATA(J) = IDATA(J)
          IF (IPT .LE. 1 ) GO TO 115
      DO 110 M = 1, IPT - 1
          SMOOTH(M) = JDATA(M)
115      DO 130 M = IPT, MAXSMO
          SMOOTH(M) = C(1) * JDATA(M)
          CSUM = C(1)
          IMAX = NC1
          IF (M .EQ. IPT) GO TO 125
          IF (M .LT. NC1 + IPT) IMAX = M - IPT
      DO 120 I = 1, IMAX
          IP1 = I + 1
          CSUM = CSUM + 2 * C(IP1)
120      SMOOTH(M) = SMOOTH(M)+C(IP1)*JDATA(M+I)+C(IP1)*JDATA(M-I)
125      SMOOTH(M) = SMOOTH(M) / CSUM
130      CONTINUE
      DO 140 I = 1, NSHIFT
140      JDATA(I) = JDATA(ISHIFT+I)
          IF (IFSEC.EQ.IISEC) GO TO 320
C
C *****
C
C BEGIN SMOOTHING LOOP:
C
```

```

DO 300 K=IISEC+1,IFSEC
      IF(10*(K/10).EQ.K) PRINT *,'THE FIRST',K,'SECTORS
$ HAVE BEEN PROCESSED THUS FAR'
      K1 = K - 1
      CALL IRTISK( IDATA, 512, ISFSEC + K, INODE )
DO 150 I = 1, 512
      IF ( IDATA(I) .LT. -524287 ) IDATA(I) = 0
150     JDATA( I + NSHIFT ) = IDATA(I)
DO 180 M = 1, NC1
      MM = MAXSMO + M
      SMOOTH( MM ) = C(1) * JDATA( NC1 + M )
      NC1PM = NC1 + M
DO 170 I = 1, NC1
      IP1 = I + 1
170     SMOOTH(MM) = SMOOTH(MM) + C(IP1) * JDATA( NC1PM + I )
$           +C(IP1) * JDATA( NC1PM - I )
180     CONTINUE
      CALL IWTISK( SMOOTH, 512, IDFSEC + K1, INODE)
DO 250 M = 1, MAXSMO
      MM = NSHIFT + M
      SMOOTH(M) = C(1) * JDATA( MM )
DO 240 I = 1, NC1
      IP1 = I + 1
240     SMOOTH(M) = SMOOTH(M) + C(IP1) * JDATA( MM + I )
$           + C(IP1) * JDATA( MM - I )
250     CONTINUE
DO 260 I = 1, NSHIFT
260     JDATA(I) = JDATA( 512 + I )
300     CONTINUE
C
C *****
C
320     DO 350 I = 1, NC1
350     SMOOTH( MAXSMO + I ) = IDATA( MAXSMO + I )
      CALL IWTISK( SMOOTH, 512, IDFSEC + IFSEC, INODE )
      CALL EXIT
      END

```

```

!
C -----
C "SORT.FOR"
C
C SUBROUTINE OF "AMAC.FOR"
C
C -----
      SUBROUTINE SORT(NDATA,DATA,INDOUT)
      DIMENSION DATA(200),INDOUT(200)
C
C THIS SUBROUTINE SORTS "NDATA" DATA PTS FROM A "DATA" ARRAY

```

```

C IN ASCENDING ORDER. "INDOUT" IS AN OUTPUT ARRAY OF INDICES
C THAT RELATES THE SORTED ORDERING TO THE UNSORTED ORDERING.
C
C INITIALIZE INDICES:
C
      DO 11 I = 1, NDATA
11      INDOUT(I) = I
          N1DATA = NDATA - 1
          NN = 0
C
C INCREMENT INDICES:
C
21      NN = NN + 1
          IF ( NN. GT . N1DATA )                GO TO 101
31      N = NN + 1
C
C COMPARE A PAIR OF POINTS:
C
41      IF ( DATA(N). LT . DATA(NN) )        GO TO 61
51      N = N + 1
          IF ( N. GT . NDATA )                GO TO 21
                                              GO TO 41

C SWAP DATA ORDERING & INDEX ORDERING FOR A PAIR OF POINTS:
C
61      TMP          = DATA(N)
          DATA(N)   = DATA(NN)
          DATA(NN)  = TMP
C
          TMP          = INDOUT(N)
          INDOUT(N)   = INDOUT(NN)
          INDOUT(NN)  = TMP
                                              GO TO 51
C
101     RETURN
        END

!
C -----
C "SPLINE.FOR"
C
C SUBROUTINE OF "BASLIN.FOR" & "UVVIS.FOR"
C -----
C SPLINE.FOR
C
C THIS SUBROUTINE COMPUTES COEFFICIENTS FOR CUBIC INTERPOLATION
C FORMULAS.
C *****
C

```

```

C INPUT:
C SPLOPT: + VALUE: CUBIC INTERPOLATION.
C 0 : SET ALL COEFFICIENTS TO 0.
C - VALUE: LINEAR INTERPOLATION.
C NPTS = # OF DATA POINTS
C XDATA(I),YDATA(I) = DATA ARRAYS
C Y2DI,Y2DF = ESTIMATES OF THE 2ND DERIVATIVES AT THE
C INITIAL AND FINAL INTERPOLATION POINTS,
C RESPECTIVELY.
C
C OUTPUT:
C C(I,J) = ARRAY OF COEFFICIENTS (J = 1,2,3)
C
C *****
C THE COEFFICIENTS FOR THE ITH CUBIC EQN (COVERING RANGE I TO
C I+1) ARE GIVEN BY:
C
C Y = C(I,3)*X**3 + C(I,2)*X**2 + C(I,1)*X + YDATA(I)
C
C THE COEFFICIENTS ARE DEFINED SO THAT XDATA(I) CORRESPONDS
C TO X=0.
C *****
C
C SUBROUTINE SPLINE(SPLOPT,NPTS,XDATA,YDATA,Y2DI,Y2DF,C)
C DIMENSION XDATA(10),YDATA(10),C(10,3),XUPDIF(2),YUPDIF(2),
1 DIFRAT(2),R(10),CY2D(10,10),Y2D(10)
C INTEGER SPLOPT
C
C NDIM=NPTS-2
C
C COMPUTE 2ND DERIVATIVES FROM AN ITERATIVE FORMULA
C DERIVED FROM BOUNDARY MATCHING CONDITIONS FOR 2ND DERIVATIVES.
C
C Y2D(1)=Y2DI
C Y2D(NPTS)=Y2DF
C MAXDIM=10
C DO 11 I=1,MAXDIM
C R(I)=0.0
C DO 11 J=1,3
11 C(I,J)=0.0
C IF (SPLOPT) 15,999,19
15 DO 17 I=1,MAXDIM
C DENOM=XDATA(I+1)-XDATA(I)
C IF (ABS(DENOM).LT.1.0E-6) GO TO 999
17 C(I,1)=(YDATA(I+1)-YDATA(I))/DENOM
C GO TO 999
19 DO 21 I=1,2
C XUPDIF(I)=XDATA(I+1)-XDATA(I)
C YUPDIF(I)=YDATA(I+1)-YDATA(I)
21 DIFRAT(I)=YUPDIF(I)/XUPDIF(I)
C R(1)=-XUPDIF(1)*Y2D(1)
C DO 51 I=1,NDIM
C IF (I.EQ.1) GO TO 31

```

```

          CY2D(I,I-1)=XUPDIF(1)
31      CY2D(I,I)=2*(XUPDIF(1)+XUPDIF(2))
          IF (I.EQ.NDIM) GO TO 41
          CY2D(I,I+1)=XUPDIF(2)
41      R(I)=6*(DIFRAT(2)-DIFRAT(1))+R(I)
          IF (I.EQ.NDIM) GO TO 51
          XUPDIF(1)=XUPDIF(2)
          XUPDIF(2)=XDATA(I+3)-XDATA(I+2)
          YUPDIF(2)=YDATA(I+3)-YDATA(I+2)
          DIFRAT(1)=DIFRAT(2)
51      DIFRAT(2)=YUPDIF(2)/XUPDIF(2)
          R(NDIM)=R(NDIM)-XUPDIF(2)*Y2D(NPTS)
C
C INVERT THE MATRIX OF 2ND-DERIVATIVE COEFFICIENTS:
C
          CALL MATINV(NDIM,CY2D)
C
C *****
C COMPUTE 2ND DERIVATIVES FOR POINTS 2,3,...NPTS-1.
C
          DO 131 I=2,NPTS-1
              Y2D(I)=0.0
          DO 131 J=1,NDIM
131      Y2D(I)=Y2D(I)+CY2D(I-1,J)*R(J,1)
C
C *****
C THE FOLLOWING EXPRESSIONS FOR THE CUBIC COEFFICIENTS ARE TRUE
C IN GENERAL BUT THE 2ND DERIVATIVES THAT APPEAR IN THESE
C EXPRESSIONS ARE DEPENDENT ON INPUT BOUNDARY CONDITIONS.
C
          DO 301 I=1,NPTS-1
              Q= XDATA(I+1)-XDATA(I)
              C(I,3)= (Y2D(I+1)-Y2D(I))/(6*Q)
              C(I,2)= Y2D(I)/2
              C(I,1)= (YDATA(I+1)-YDATA(I)-Q*Q*(Y2D(I)/3+Y2D(I+1)/6))/Q
301      CONTINUE
999      RETURN
          END

```

```

!
C -----
C "STICK.FOR"
C
C CALLS: FTIR SYSTEM SUBROUTINES
C
C CALLING MACRO: STK
C
C -----

```

```

C
C      PROGRAM STICK
C
C "STICK.FCP" GENERATES A STICK SPECTRUM USING INPUT
C ABSORBANCE DATA AT SEVERAL DISCRETE CM-1 VALUES. IT IS RUN
C FROM MACRO "STK". THE INPUT DATA MUST BE ENTERED INTO FILE
C "STICK.DAT" AS FOLLOWS:
C
C 1) ENTER THE # OF INPUT DATA POINTS (I4 FORMAT).
C
C 2) ENTER A PAIR OF (CM-1, ABSORBANCE) VALUES ON EACH NEW
C LINE (2F7 FORMAT).
C
C 3) REPEAT STEPS 1 AND 2 FOR ANY ADDITIONAL SPECTRA.
C
C      DIMENSION IDATA(512),F(50),A(50)
C      OPEN(12,FILE='STICK.DAT',FORM='FORMATTED',STATUS='OLD')
C
C*****
C READ FTIR PARAMETERS:
C
C      IDFN=DFN      NSECS~FSZ      ID=VIO (DATA SET #)
C
C      INODE= IRVAL(13004,0)
C      ID= IRVAL(13762,0)
C      NSECS= IRVAL(14001,0)
C      NPTS= IRVAL(14002,0)*128
C      IDFN= IRVAL(14022,0)
C*****
C
C LOCATE SECTORS TO BE READ USING THE FOLLOWING INDEX:
C
C      IDFSEC = IDFN * NSECS + 88 - 1
C
C SET EXP=1 FOR THE TRANSMITTANCE FILE:
C
C      CALL IRTISK( IDATA, 512, IDFSEC + NSECS, INODE )
C      IDATA(6) = 1
C      CALL IWTISK( IDATA, 512, IDFSEC + NSECS, INODE )
C
C WRITE TRANSMITTANCE = 1.0 FOR ALL DATA POINTS IN THE SCRATCH
C FILE TO CREATE A BASELINE (NOTE: DATA WORD = 262144 FOR T=1):
C
C      PPWVN = 4.148226 * NPTS / 32768
C      MAXSEC = NPTS / 512
C      DO 50 J = 1, 512
50      IDATA(J) = 262144
C      DO 100 I = 1, MAXSEC
100     CALL IWTISK( IDATA, 512, IDFSEC + I, INODE )
C *****
C
C READ INPUT DATA FROM FILE "STICK.DAT":
C

```

```

DO 160 K = 1, ID
    READ(12,5000) NLINES
    PRINT *,NLINES
DO 150 J = 1, NLINES
    READ(12,4000) F(J), A(J)
150    PRINT *,F(J), A(J)
160    CONTINUE
4000  FORMAT(2F7.2)
5000  FORMAT(I4)
C
C*****
C
C SPECTRUM PROCESSING LOOP:
C
    DO 200 I = 1, NLINES
        IWORD = 262144 * 10**( -A(I) )
        N = 1 + PPWVN * F(I)
        ISEC = N / 512 + 1
        NLOCAL = N - 512 * ( ISEC - 1 )
        CALL IRTISK( IDATA, 512, IDFSEC + ISEC, INODE )
        IDATA( NLOCAL ) = IWORD
200    CALL IWTISK( IDATA, 512, IDFSEC + ISEC, INODE )
C
C*****
        CALL EXIT
        END

!
C -----
C "SUMSPC.FOR"
C
C CALLS: 1) "FTPARM.FOR"
C         2) FTIR SYSTEM SUBROUTINES
C
C CALLING MACRO: SUM
C -----
C
C PROGRAM SUMSPC
C
C THIS PROGRAM TAKES A LINEAR COMBINATION OF FTIR SCRATCH FILES
C ("IRFN" & "IDFN") IN THE CM-1 RANGE FROM "RFXF" TO "RLXF", AND
C SENDS THE RESULTS TO FILE "OFN"; I.E.
C
C FCR * (FILE IRFN) + FCD * (FILE IDFN) + ADDCON = FILE OFN
C
C WHERE "FCR", "FCD", & "ADDCON" ARE CONSTANTS. THIS PROGRAM IS
C SIMILAR TO THE FTIR "ADD" OPERATION EXCEPT THAT:
C

```



```

C      1) THE OUTPUT MAY GO TO A NEW FILE.
C      2) ANY TYPE OF SPECTRA CAN BE ADDED.
C      3) AN ADDITIVE CONSTANT "ADDCON" CAN BE USED.
C
C BEFORE EXECUTING THIS PROGRAM, THE INPUT FILES MUST BE
C AUTOSCALED (C.F. MACRO "SU0") & THE MAXIMA MUST BE STORED IN
C FTIR PARAMETERS VF1 (YRMAX) & VF2 (YDMAX).
C
C WHEN ICHEXP=999 (FTIR PARAMETER 'SIZ') THE EXPONENT OF THE
C OUTPUT FILE IS SET AT 4; I.E. THE EXPONENT-DETERMINATION
C ALGORITHM IS SKIPPED. THIS OPTION IS USED BY MACRO 'VCR'
C BECAUSE PIECEWISE ADDITION IS EMPLOYED. NOTE THAT FOR 'VCR',
C THE OUTPUT FILE # MUST DIFFER FROM THE INPUT FILE #'S
C BECAUSE OTHERWISE CHANGING THE INPUT FILE EXPONENT TO 4 WOULD
C ALTER THE ENTIRE FILE BEFORE THE FILE WAS FULLY PROCESSED.
C
C WHEN ICHEXP=9999 AND IDFN=OFN, THE EXPONENT OF OFN IS SET TO
C WHATEVER THE EXPONENT OF IDFN IS.
C
C      INTEGER   OFN, OFSEC, OEXP
C      DIMENSION IRDATA(512), IDDATA(512), IODATA(512)
C
C READ FTIR PARAMETERS AND INITIALIZE SECTOR INDICES:
C
C      INODE = IRVAL(13004,0)
C      ICHEXP = IRVAL(13714,0)
C      YRMAX = RVAL(13770,0)
C      YDMAX = RVAL(13772,0)
C      ADDCON = RVAL(13774,0)
C      NSECS = IRVAL(14001,0)
C      IDFN = IRVAL(14022,0)
C      OFN = IRVAL(14023,0)
C      IRFN = IRVAL(14030,0)
C      FCD = RVAL(14062,0)
C      FCR = RVAL(14066,0)
C      RFXF = RVAL(14135,0)
C      RLXF = RVAL(14213,0)
C
C      IRFSEC = IRFN * NSECS + 88 - 1
C      IDFSEC = IDFN * NSECS + 88 - 1
C      OFSEC = OFN * NSECS + 88 - 1
C
C READ THE FILE STATUS BLOCKS OF THE INPUT FILES TO:
C
C 1) DETERMINE THE HENE CM-1.
C 1) IDENTIFY SPECTRA AS 1-BEAM, TRANSMITTANCE, OR ABSORBANCE.
C 2) DETERMINE THE EXPONENT AND # OF SCANS FOR THE SPECTRA.
C
C      CALL IRTISK( IRDATA, 512, IRFSEC + NSECS, INODE )
C      IREXP = IRDATA(6)
C      CALL IRTISK( IDDATA, 512, IDFSEC + NSECS, INODE )
C      IDEXP = IDDATA(6)
C      ISNGLE = IDDATA(7)

```

```

      IHENE = IDDATA(333)
C
C SET DATA BOUNDARY PARAMETERS:
C
      CALL FTPARM( IHENE, RFXF, RLXF, FROMIN, FROMAX, IISEC,
1             IFSEC, INDXSP, INDEXP, PPWVN )
C
C FOR 1-BEAM SPECTRA, THE #'S OF SCANS OF THE 2 INPUT FILES ARE
C ADDED SINCE THE WEIGHTING FACTORS ARE ALREADY INCLUDED IN THE
C STORED DATA VALUES. FOR OTHER SPECTRA, THE #'S OF SCANS ARE
C IRRELEVANT.
C
      IF (ISNGLE.EQ.0) IDDATA(3) = IRDATA(3) + IDDATA(3)
C
C USE THE MACRO AUTOSCALING RESULTS TO DETERMINE THE EXPONENT
C FOR THE OUTPUT FILE. THEN SEND THE REVISED FILE STATUS BLOCK
C OF THE INPUT FILE "IRFN" TO THE OUTPUT FILE "OFN".
C
      NX      = -524288
      HX      = 524288.0
      RSCALE = 2 ** ( 19 - IREXP )
      DSCALE = 2 ** ( 19 - IDEXP )
      IF ( ICHEXP.EQ. 999 ) THEN
          OEXP = 4
      ELSEIF (ICHEXP.EQ.9999. AND. IDFN.EQ.OFN) THEN
          OEXP = IDEXP
      ELSE
          OEXP = 0
          SUMMAX = (FCR*YRMAX +FCD*YDMAX +ADDCON) * HX
DO 51      I = 1, 100
          IF ( SUMMAX. LT. HX )                      GO TO 81
          SUMMAX = SUMMAX / 2
51         OEXP = OEXP + 1
          PRINT *, 'EXPONENT ERROR IN SUM.FOR'
          IF ( 2. EQ. 2 )                          GO TO 999
      ENDIF
81         IDDATA(6) = OEXP
          CALL IWTISK( IDDATA, 512, OFSEC + NSECS, INODE )
C
C READ THE 2 INPUT FILES, ADD THEM, AND SEND THE RESULTS TO THE
C OUTPUT FILE. A BLANKED INPUT WORD IS CONVERTED TO ZERO
C WHENEVER THE CORRESPONDING WORD IN THE OTHER INPUT SPECTRUM IS
C NOT BLANKED.
C
          OSCALE = 2 ** ( 19 - OEXP )
DO 201 M = IISEC, IFSEC
          CALL IRTISK( IRDATA, 512, IRFSEC + M, INODE )
          CALL IRTISK( IDDATA, 512, IDFSEC + M, INODE )
          CALL IRTISK( IODATA, 512, OFSEC + M, INODE )
DO 101 I = 1, 512
          KEEP1 = 1
          KEEP2 = 1
          IF ( IRDATA(I). EQ. NX )          PR = 0

```

```

IF ( IDDATA(I). EQ. NX )   KEEPD = 0
IF ( KEEPDR. EQ. 0. AND. KEEPDR. EQ. 0 )   GO TO 101
IF ( M. EQ. IISEC. AND. I. LT. INDXSP )   GO TO 101
IF ( M. EQ. IFSEC. AND. I. GT. INDXEP )   GO TO 101
IODATA(I) = ( KEEPDR * FCD * IDDATA(I) / DSCALE +
1          KEEPDR*FCR*IRDATA(I)/RSCALE)*OSCALE+ADDCON
101      CONTINUE
201      CALL IWTISK( IODATA, 512, OFSEC + M , INODE )
999      CALL EXIT
          END

```

!

```

C -----
C "UVVIS.FOR"
C
C CALLS: 1) "FSBTIT.FOR"
C         2) "SPLINE.FOR" > "MATINV.FOR"
C         3) FTIR SYSTEM SUBROUTINES
C
C CALLING MACRO: UVV
C -----
C
C          PROGRAM UVVIS
C
C "UVVIS.FCP" COPIES UV-VISIBLE DATA FROM A NICOS FILE INTO FTIR
C SCRATCH FILE "DFN". TO RUN THIS PROGRAM, THE FOLLOWING
C PRELIMINARY STEPS MUST BE ACCOMPLISHED:
C
C     1) SEND SURVEY SPECTRA (474 PTS PER FILE) FROM THE PE
C         DIODE ARRAY SPECTROMETER TO THE HP150 USING THE
C         "SND.OY" PROGRAM ON THE PE DATA STATION, AND A LOGGING
C         FILE ON THE 50 OR ZENITH.
C
C     2) ARRANGE THE DATA IN COLUMNS USING PDAPAK.FOR ON THE
C         ZENITH OR HP150.
C
C     3) COPY THE ZENITH OR HP150 FILE TO THE NICOLET USING
C         XMODEM.
C
C          INTEGER  TITLE(88)
C          DIMENSION IDATA(512),WVL(474),WVLSPL(10),ASPL(10),
C          $          CD(10,3),A(7,474)
C          REAL     NI
C          CHARACTER TERM*80, CONC*80, LET*2, LETSTR*80, LETNEW*3,
C          $          FILNAM*80,TRIM*80
C*****
C
C INITIALIZE PARAMETERS:
C

```

```

                PRINT *, 'UVVIS.FCP EXECUTION UNDERWAY'
                NPEPTS = 474
                IEXP   = 2
                NTP    = 2 * 512
                WVLII  = 900
                WVLFI  = 190
                WVLPPI = ( WVLII - WVLFI ) / ( NPEPTS - 1 )
                WVLPP0 = ( WVLII - WVLFI ) / ( NTP - 1 )
DO 11 I = 1, 4
11      WVLSPL(I) = I
C
C*****
C READ FTIR PARAMETERS:
C
        INODE = IRVAL(13004,0)
        NSECS = IRVAL(14001,0)
        IDFN  = IRVAL(14022,0)
        IQIT  = IRVAL(14222,0)
C
C *****
C READ DATA FROM INPUT FILE:
C
        OPEN( 12, FILE='UVVIS.DAT[UVV]', FORM='FORMATTED',
$ STATUS='OLD' )
        IF (IQIT.EQ.1) THEN
                READ(12,100) WVL(1), A(1,1)
        ELSEIF (IQIT.EQ.2) THEN
                READ(12,100) WVL(1), A(1,1), A(2,1)
        ELSEIF (IQIT.EQ.3) THEN
                READ(12,100) WVL(1), A(1,1), A(2,1), A(3,1)
        ELSEIF (IQIT.EQ.4) THEN
                READ(12,100) WVL(1), A(1,1), A(2,1), A(3,1), A(4,1)
        ELSEIF (IQIT.EQ.5) THEN
                READ(12,100) WVL(1), A(1,1), A(2,1), A(3,1), A(4,1)
$                ,A(5,1)
        ELSEIF (IQIT.EQ.6) THEN
                READ(12,100) WVL(1), A(1,1), A(2,1), A(3,1), A(4,1)
$                ,A(5,1), A(6,1)
        ELSEIF (IQIT.EQ.7) THEN
                READ(12,100) WVL(1), A(1,1), A(2,1), A(3,1), A(4,1)
$                ,A(5,1), A(6,1), A(7,1)
        ENDIF
C
DO 101 I = 2, NPEPTS
IF (IQIT.EQ.1) THEN
        READ(12,100) WVL(I), A(1,I)
ELSEIF (IQIT.EQ.2) THEN
        READ(12,100) WVL(I), A(1,I), A(2,I)
ELSEIF (IQIT.EQ.3) THEN
        READ(12,100) WVL(I), A(1,I), A(2,I), A(3,I)
ELSEIF (IQIT.EQ.4) THEN
        READ(12,100) WVL(I), A(1,I), A(2,I), A(3,I), A(4,I)

```

```

        ELSEIF (IQIT.EQ.5) THEN
            READ(12,100) WVL(I), A(1,I), A(2,I), A(3,I), A(4,I)
$           ,A(5,I)
        ELSEIF (IQIT.EQ.6) THEN
            READ(12,100) WVL(I), A(1,I), A(2,I), A(3,I), A(4,I)
$           ,A(5,I), A(6,I)
        ELSEIF (IQIT.EQ.7) THEN
            READ(12,100) WVL(I), A(1,I), A(2,I), A(3,I), A(4,I)
$           ,A(5,I), A(6,I), A(7,I)
        ENDIF
101  CONTINUE
100  FORMAT( 1X, F6.2, 7(1X, F6.4) )
C
C*****
C
    DO 301 J = 1, IQIT
        IFILE = IDFN + J - 1
        PRINT *, 'SENDING DATA TO FILE #: ', IFILE
        IDFSEC = IFILE * NSECS + 87
C
C SET OUTPUT FILE PARAMETERS:
C 1) 16 CM-1 RESOLUTION (DETERMINED BY "NTP")
C 2) 190 NM TO 900 NM
C 3) FILE EXPONENT = IEXP
C
    CALL IRTISK( IDATA, 512, IDFSEC + NSECS, INODE )
    IDATA(6) = IEXP
    IDATA(7) = -1
    IDATA(8) = -1
    IDATA(9) = -1
    IDATA(10) = 0
    IDATA(11) = 0
    IDATA(15) = 8
    IDATA(16) = 8
    IDATA(24) = 190
    IDATA(25) = 900
    CALL IWTISK( IDATA, 512, IDFSEC + NSECS, INODE )
C
C *****
C
C CONVERT DATA TO NICOLET'S STYLE OF DATA STORAGE. "NI" & "NO"
C ARE THE ORDERING INDICES FOR THE PERKIN-ELMER & NICOLET DATA,
C RESPECTIVELY. THE # OF NICOLET DATA PTS IS ARBITRARILY SET
C AT NTP, REGARDLESS OF THE # OF PERKIN-ELMER PTS.
C INTERPOLATION IS USED TO CONVERT FROM THE INPUT DATA INTERVAL
C TO THE OUTPUT DATA INTERVAL.
C
        IDATA(1) = A(J, NPEPTS) * 2**( 19-IEXP )
    DO 201 NO = 2, NTP
        NI = 1 + ( ( 1 - NO ) * WVLPP0 + 710 ) / WVLPP1
        IILOW = INT(NI)
        REM = NI - IILOW
C

```

```

IF ( ILOW + 2 . LE. NPEPTS .AND. ILOW.GT.1 ) THEN
  ASPL(1) = A(J, ILOW-1)
  ASPL(2) = A(J, ILOW )
  ASPL(3) = A(J, ILOW+1)
  ASPL(4) = A(J, ILOW+2)
  CALL SPLINE(1,4,WVLSPL,ASPL, 0.0, 0.0, CD)
  AO = A(J,ILOW) + CD(2,1)*REM +CD(2,2)*REM*REM
  $      + CD(2,3)*REM*REM*REM
      ELSE
  $      AO = A(J,ILOW)
      + REM * ( A(J, ILOW+1) - A(J, ILOW) )
      ENDIF
C
ISEC      = ( NO - 1 ) / 512
I         = NO - 512 * ISEC
IDATA(I) = AO * 2**( 19 - IEXP )
ISEC      = IDFSEC + ISEC + 1
201      IF (I.EQ.512) CALL IWTISK(IDATA,512,ISEC,INODE)
301      CONTINUE
C
C*****
999      CALL EXIT
      END

```

APPENDIX B

COMPUTER PROGRAMS FOR THE PERKIN-ELMER 7500 WORK STATION

The following software enables a block of UV-VIS spectral data files to be sent from the PE-7500 work station of the PE-3840 UV-VIS spectrometer to a logging file on an HP-150 or a Zenith-248 computer. Further transfer of data can be made from an HP-150 or Zenith-248 computer to the work station of the Nicolet Model 740 FT-IR spectrometer using XMODEM. Several benefits accrue from this procedure:

1. Data manipulations can be performed with IBM-compatible software (files created on the Zenith-248 can be used in this manner).
2. Plots can be produced with the GRAFIT software of the HP-150.
3. Plotting and other data manipulations can be performed using Nicolet's software. UV-VIS data can be loaded into Nicolet's FT-IR scratch files using the UVV macro described in Appendix A.
4. The speed of data manipulation can be greatly increased.

```
-----  
* "snd.oy"          (OBEY program)  
*  
* calls: "hp.ba"  
* -----  
*  
* "snd.oy" sends a block of "*.sp" spectral data files from  
* a floppy disk on the Perkin-Elmer 7500 to a logging file on an  
* HP-150 or Zenith-248 via an intermediate PE-7500 scratch file  
* "usr/uv/data/i.da". All of the input files should be sent to  
* one receiving file called "pda.dat" where they can later be  
* re-separated into spectral files using "pdastk.for".  
&ll *  
* The input files must have:  
*  
* 1) a common file name of the form:          GEN###.sp  
*      where GEN is a generic name consisting of 2 to 5  
*      alphanumeric characters.  
*  
* 2) a data range from 190nm to 900nm  
*  
* Enter the # of files of input spectral data:
```

```

&enter a1
*
* Enter a data point reduction factor for RS-232 data
* transmission:
*     1 = send every spectral point.
*     2 = send every other spectral point.
*     etc.
*
&enter a10
*
* Enter all except the last 5 characters of the GENERIC input
* file name; e.g. if the first file is named blank001.sp,
* enter: bla
&enter a2
*
* Enter the last 5 characters of the GENERIC input file name; the
* last 3 characters must be a 3-digit identification for the
* initial file in the set (ignore ".sp"). In the above example,
* one would enter: nk001
&enter a3
*
* 1) PUT THE DISK CONTAINING THE SPECTRAL DATA INTO DRIVE 0.
* A SECOND DATA DISK MAY BE INSERTED INTO DRIVE 1 IF THE
* FILE EXTENSIONS CONTINUE SEQUENTIALLY ONTO THIS SECOND
* DISK.
* 2) SET UP RECEIVING FILE "A:PDA.DAT" ON THE RECEIVING
* COMPUTER.
* 3) PRESS [ENTER] TO BEGIN DATA CONVERSION. TO ABORT THIS
* PROGRAM, PRESS [BREAK] AND THEN TYPE "VSAVE CLOSE".
*
do pause
do display off
calc v64=&a1
&def a17= "w4:i"
&def a6= "f0:"
&def a7= "f1:"
&def a8= a6
&for v2=1,v64
&def a4= a2 + a3
&def a9= a8 + a4 + ".sp"
do display on
&check a9
&error 160
&goto 162
&l60
&def a8=a7
&def a9=a8 + a4 + ".sp"
&check a9
&error 161
&goto 162
&l61 *FILE NOT FOUND*
&goto 11
&l62 retrieve x &a9

```



```

calc v1=xstrt+0.001
calc v5=&a10*xdel
calc v3=int(xnpts/&a10+0.001)
calc v6=int(v3/20)
calc v7=int(-20*v6+v3)
vsave open new &a17
vsave &a17 v2
vsave &a17 a9
do display off
vsave &a17 v1
vsave &a17 v5
vsave &a17 v3
calc v10=xstrt+v5
&for v60=1,v6
calc v10=v10-v5
calc v11=x(v10)
calc v10=v10-v5
calc v12=x(v10)
calc v10=v10-v5
calc v13=x(v10)
calc v10=v10-v5
calc v14=x(v10)
calc v10=v10-v5
calc v15=x(v10)
calc v10=v10-v5
calc v16=x(v10)
calc v10=v10-v5
calc v17=x(v10)
calc v10=v10-v5
calc v18=x(v10)
calc v10=v10-v5
calc v19=x(v10)
calc v10=v10-v5
calc v20=x(v10)
calc v10=v10-v5
calc v21=x(v10)
calc v10=v10-v5
calc v22=x(v10)
calc v10=v10-v5
calc v23=x(v10)
calc v10=v10-v5
calc v24=x(v10)
calc v10=v10-v5
calc v25=x(v10)
calc v10=v10-v5
calc v26=x(v10)
calc v10=v10-v5
calc v27=x(v10)
calc v10=v10-v5
calc v28=x(v10)
calc v10=v10-v5
calc v29=x(v10)
calc v10=v10-v5

```

```

calc v30=x(v10)
vsave &a17 v11
vsave &a17 v12
vsave &a17 v13
vsave &a17 v14
vsave &a17 v15
vsave &a17 v16
vsave &a17 v17
vsave &a17 v18
vsave &a17 v19
vsave &a17 v20
vsave &a17 v21
vsave &a17 v22
vsave &a17 v23
vsave &a17 v24
vsave &a17 v25
vsave &a17 v26
vsave &a17 v27
vsave &a17 v28
vsave &a17 v29
vsave &a17 v30
&next v60
&for v8=1,v7
calc v10=v10-v5
calc v11=x(v10)
vsave &a17 v11
&next v8
vsave close
idris basic -r /usr/uv/data/hp.ba
&incr a3
&next v2
* ALL FILES CONVERTED
&ll1000 vsave close
* END

```

"hp.ba" (BASIC program)

Subroutine of "snd.oy"

```

$10 rem: "hp.ba" is a subroutine that reads spectral data from
20 rem: file "/usr/uv/data/i.da" on the Perkin-Elmer 7500, and
30 rem: sends it to a logging file on an HP-150 or a Zenith-248
40 rem: via the tty0 port.
50 dim s(10)
60 defwrn n
70 fi$="/usr/uv/data/i.da"
80 fo$="/dev/tty0"
90 open "i",# 5,fi$

```

```

100 open "o", # 6, fo$
110 input # 5, junk
120 input # 5, iter
130 input # 5, junk
140 input # 5, fhp$
150 input # 5, junk
160 input # 5, wvli
170 input # 5, junk
180 input # 5, wvlspa
190 input # 5, junk
200 input # 5, ndp
210 rdp= 1.0 *ndp
220 n= len (fhp$)
230 ofil$=fhp$
240 nxt1= asc ( right$ (fhp$, 3))
250 nxt2= asc ( right$ (fhp$, 2))
260 nxt3= asc ( right$ (fhp$, 1))
270 rem: check that last 3 characters of file name are digits
280 if nxt1< 48 or nxt1> 57 goto 400
290 if nxt2< 48 or nxt2> 57 goto 400
300 if nxt3< 48 or nxt3> 57 goto 400
310 nxt= 100*nxt1+ 10*nxt2+nxt3- 5329+iter
320 ndig1=nxt/100
330 sbt=nxt- 100*ndig1
340 ndig2=sbt/ 10
350 ndig3=sbt-10*ndig2
360 ndig1=ndig1+ 48
370 ndig2=ndig2+ 48
380 ndig3=ndig3+ 48
390 ofil$=left$(fhp$,n- 3)+ chr$(ndig1)+ chr$(ndig2)+ chr$(ndig3)
400 print # 6, ofil$
410 print # 6, wvli
420 print # 6, wvlspa
430 print # 6, rdp
440 npline= 8
450 nfull=ndp/npline
460 for i=1 to nfull
470 for j=1 to npline
480 input # 5, junk, s(j)
490 next j
500print#6, using "-#.####"; 1);s(2);s(3);s(4);s(5);s(6);s(7);s(8)
510 next i
520 nrem=ndp-npline*nfull
530 if nrem= 0 goto 610
540 for i=1 to nrem
550 input # 5, junk, s(i)
560 next i
570 for i=nrem+1 to npline
580 s(i)= 0
590 next i
600print#6, using "-#.####";s(1);s(2);s(3);s(4);s(5);s(6);s(7);s(8)
610 system

```

"PDAPAK.FOR"

Calls: none

PROGRAM PDAPAK

C
C "PDAPAK.FOR" READS UV-VIS SURVEY SPECTRAL DATA (474 PTS / FILE)
C WHICH HAS BEEN SENT TO AN HP-150 OR ZENITH-248 LOGGING FILE
C NAMED "PDA.DAT" BY PERKIN-ELMER 7500 PROGRAM "SND.OY".
C IT ORGANIZES THE DATA SO THAT EACH ROW CORRESPONDS TO A
C DIFFERENT WAVELENGTH, AND EACH COLUMN CORRESPONDS TO A
C DIFFERENT SPECTRUM. EIGHT COLUMNS ARE WRITTEN TO OUTPUT FILE
C PDASTK.DAT (MAX OF 7 SPECTRA).
C

```
DIMENSION Y(3324)
CHARACTER FILNAM*10
WRITE(*,400)
      OPEN(11,FILE='PDAPAK.DAT',STATUS='UNKNOWN',FORM=
:         'FORMATTED')
      OPEN(10,FILE='PDA.DAT',STATUS='OLD',FORM='FORMATTED')
DO 201 NFIL=1,7
      READ(10,100,END=301) FILNAM
      WRITE(*,600) FILNAM
      IF (NFIL.EQ.1) THEN
        WRITE(11,700) FILNAM
        WRITE(11,800)
        WRITE(11,900)
        WRITE(11,800)
      ENDIF
      N = NFIL
      READ( 10, 200 ) WVLI
      READ( 10, 200 ) WVLSPA
      READ(10,200) RDP
      WRITE(*,200) RDP
      NPLINE = 8
      NDP     = NINT(RDP)
      NLINES = NDP / NPLINE
      NREM   = NDP - NPLINE * NLINES
      IMAX  = NLINES
      IF ( NREM .EQ. 0 ) IMAX = NLINES - 1
      L = ( NFIL - 1 ) * NDP + 1
DO 101 I = 0, IMAX
```

```

1          READ(10, 200) Y(L), Y(L+1), Y(L+2), Y(L+3), Y(L+4),
101          Y(L+5), Y(L+6), Y(L+7)
201          L = L + 8
301          CONTINUE
          N2 = 2 * NDP
          N3 = 3 * NDP
          N4 = 4 * NDP
          N5 = 5 * NDP
          N6 = 6 * NDP
          W0 = WVLI + WVLSPA
DO 311     J = 1, NDP
          WVL = W0 - J * WVLSPA
          IF ( N .EQ. 1 ) THEN
              WRITE(11,300) WVL, Y(J)
          ELSEIF ( N .EQ. 2 ) THEN
              WRITE(11,300) WVL, Y(J), Y(J+NDP)
          ELSEIF ( N .EQ. 3 ) THEN
              WRITE(11,300) WVL, Y(J), Y(J+NDP), Y(J+N2)
          ELSEIF ( N .EQ. 4 ) THEN
              WRITE(11,300) WVL, Y(J), Y(J+NDP), Y(J+N2), Y(J+N3)
          ELSEIF ( N .EQ. 5 ) THEN
              WRITE(11,300) WVL, Y(J), Y(J+NDP), Y(J+N2), Y(J+N3),
$              Y(J+N4)
          ELSEIF ( N .EQ. 6 ) THEN
              WRITE(11,300) WVL, Y(J), Y(J+NDP), Y(J+N2), Y(J+N3),
$              Y(J+N4), Y(J+N5)
          ELSEIF ( N .EQ. 7 ) THEN
              WRITE(11,300) WVL, Y(J), Y(J+NDP), Y(J+N2), Y(J+N3),
$              Y(J+N4), Y(J+N5), Y(J+N6)
          ENDIF
311         CONTINUE
C
100         FORMAT(A10)
200         FORMAT(8F8.0)
300         FORMAT(1X, F6.2, 7(1X, F6.4))
400         FORMAT(1X, 'EXECUTION UNDERWAY')
500         FORMAT(1X, I3, ' SPECTRA WERE PROCESSED')
600         FORMAT(1X, A10)
700         FORMAT('NAME OF FIRST FILE (COLUMN #2): ', A10)
800         FORMAT(1X, ' ')
900         FORMAT(4X, 'NM      001      002      003      004',
$              '      005      006      007')
C
9999        WRITE(*, 500) N
          STOP
          END

```