FTIR-ATR CHARACTERIZATION

OF AQUIFER MINERALS

By

KAILASH SWARNA

Bachelor of Science

University of Madras

Madras, India

1983

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the degree of
MASTER OF SCIENCE
July, 1988

in memory of my father

FTIR-ATR CHARACTERIZATION OF AQUIFER MINERALS

Thesis Approved:

M G Rockley
Thesis Adviser

J Paul Devlin

N E Purdie

Norman N. Durham
Dean of the Graduate College

ACKNOWLEDGEMENTS

I would like to express my thanks to the following people who served as members of my advisory committee: Drs. M. G. Rockley, Neil Purdie, and J. P. Devlin. I am particularly indebted to Dr. M. G. Rockley for having shared with me his valuable and extensive knowledge about chemistry and many other subjects and having prepared me for an advanced degree. I also wish to thank him for his infinite patience in dealing with many of my problems. His gentle manner, kindness and humor will always continue to inspire me. Special thanks are due to Dr. Natalie Rockley for all the time and effort she has spent in showing me the workings of this wonderful lab. Without her help this thesis would not be in its present form.

I sincerely appreciate support from the department of chemistry in the form of teaching and research assistantships. Financial support from various other grants is also gratefully acknowledged.

I wish to thank all my friends in the chemistry department who made my stay here very special. I would particularly like to thank Mark Fisher for his help and friendship.

I am thankful to my parents for having provided me with the opportunity to pursue higher education and for their love.

Last, but most definitely not the least, I am grateful to my dear Sumathi, who has been a constant companion and a source of inspiration, and whose love has kept me going even in the most difficult of times.

TABLE OF CONTENTS

Chapter Page

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

The study of the interaction of organic compounds with soils is of increasing relevance as the water tables become contaminated at ever rising levels with pesticides, herbicides, hydrocarbons and other products and wastes of a society largely dependent on chemicals for many aspects of everyday life. In spite of the importance of this problem, soil minerology is a relatively new scientific discipline (1,2).

The interaction of organic compounds with soils and aquifer minerals has been known for a very long time, in a non-scientific manner. For example, the fulling process, which is used to remove grease from raw wool uses an aqueous slurry of clays. This process dates back to Biblical times (3).

Only recently has it become clear that a better understanding of clay - organic interactions is indispensible to the preservation of the ecosystem against organic pollutants. With the dramatic increase in the use of fossil fuels, the concern for the environmental effects of the pollutants produced at various stages of the production process and subsequent use of fossil fuels, particularly coal, is very real.

Sources of Pollution

Coal is composed of a great variety of polycyclic aromatic, heterocyclic aromatic, phenolic, amine, quinone, sulfur, nitrogen and other compounds, all of which have been shown to have adverse environmental effects (4-7). The production of these various fossil fuels and their conversion to useable forms of energy result in the release of very large aqueous and gaseous effluent streams (8 loc.cit). Pollution from sources which are not as obvious as energy production and consumption include large scale leaks from gasoline storage tanks, industrial solvent leakage etc.. For example, large aircraft mainteinance facilities routinely face serious leaks of paint stripping solvents. Such leaks and effluent streams, in turn release significant quantities of organic pollutants into the environment. This results in the exposure of soils and minerals to the organic pollutants, and their subsequent sorption (9).

Extent of Sorption

The extent of sorption of these organic pollutants by soils and aquifer minerals is dependent upon the nature of both the sorbent (soil or mineral) and the sorbate (pollutant).

Previous Studies of Pollutant Transport

Extensive studies in recent years have provided some understanding of the sorption of organic pollutants on soils and aquifer minerals. Mackay et al.(10) conducted a large scale field experiment on natural gradient transport of solutes in groundwater at a site in Borden, Ontario, and provided a quantitative insight into the transport and fate of five organic pollutants, based on a comparision between two inorganic tracers (chloride and bromide) and five halogenated organic chemicals (bromoform, carbon tetrachloride, tetrachloroethylene, 1,2-dichlorobenzene, and hexachloroethane). Roberts et al. (11) examined solute transport in a sand aquifer under natural gradient conditions. This work provided an extensive set of data to aid in modelling the calculations of organic solute transport in groundwater. Inglis and coworkers (12) analysed pollutant transport in a shallow unconfined aquifer and concluded on the basis of experimental results that the best solution to long range aquifer protection is the prevention of contamination. As a result of these and other studies, there has emerged a pattern that may describe the parameters responsible for pollutant sorption on aquifer minerals.

Octanol/Water Partition Coefficient

It has been established that three parameters govern the abiotic sorption of organic compounds by aquifer minerals. These are as follows :

1. Octanol-Water partition coefficient or Kow of the sorbate,(which is inversely related to compound solubility in water);

Low Kow; indicating that the organic compound is highly water soluble and that the introduction of this pollutant into the aquifer mineral or soil will result in its transport properties being dominated by the motion of the water itself or of aquifer flow rate (13).

High Kow; indicating that the nonpolar organic compound has little affinity for the aqueous phase, and thus the substrate of the aquifer determines the transport properties of the sorbate.

Total Organic Carbon Content (TOC)

2. Total Organic Carbon or TOC value of the sorbent;

High TOC: Soils / sediments with a high TOC value retain organics by a mechanism that relates primarily to the organic carbon character of the aquifer mineral (14).

Low TOC: The information available on the transport properties of nonpolar organic pollutants through low TOC soils/sediments is scant. However, it is known that low TOC aquifers predominate in pollutant plume migration pathways. TOC content of the subsurface appears to diminish with depth below the soil zone (14,15,19).

3. Sorptive capacity of clay mineral surface: In sorbents with a very low TOC content, the mineral surfaces of the soil are very important in determining the sorption properties of nonpolar organic pollutants (11,14,15,18). The surface available for sorption will vary with mineral content and with the exposed surface area. Previously published studies contain several models describing the relationship between the first two (17). While the first parameter is a property of the sorbate, the other two relate to the sorbent (18,19).

The interaction between the organic pollutants and the surface of the solid phase in low TOC aquifer minerals in the absence of biotic transformation processes is largely controlled by thermodynamic considerations. The change in the chemical potential, and the thermodynamic equilibrium constant for the sorption process (Kp), are defined mainly by the entropy of the water, the entropy of the water-solvent system, the enthalpy of the water-pollutant cage complex and the enthalpy of the pollutant-sorbent complex, and while water is the constant factor in the system, the pollutant and sorbent surface are not. The interaction of the pollutant with the water and sorbent surface will be governed by the molecular size and charge distribution. It has been proposed that this water - pollutant interaction may be adequately represented by the parameter Kow (octanol-water partition coefficient). The interaction of the sorbent with the water and the pollutant, on the other hand, is governed by the surface functional groups of the sorbent. Infrared spectroscopy as applied to the analysis of these surface functional groups, is a useful tool and is constantly being improved in its ability to analyze more complex systems. Thus, it may be possible to reliably predict the Kp values of the aquifer minerals using the data obtained from infrared spectroscopy and the Kow values of various pollutants (19).

Rockley et al. (19) have suggested that for low TOC aquifer minerals the chemical functionality of the clay mineral surface may be one of the dominant factors affecting pollutant migration. The problem was to design a suitable and reliable method to explore the existence of such a relationship. Using FTIR spectroscopy, Rockley and coworkers studied ten aquifer minerals and concluded that a relationship between factor-analyzed FTIR transmission spectra and aquifer material sorption potential did exist, at least for the limited set of sorption Kow values available (20).

X-Ray Diffractometry of Clay Minerals

This discovery showed that the use of infrared spectroscopy could indeed be very significant in analysing aquifer minerals to obtain data that could be linked to pollutant sorption. This was further substantiated by the fact that infrared spectroscopy has been used by many to study clay minerals to obtain qualitative information about several different types of clays. Of the many new and improved methods of analysis available to the chemist today, none has had a more profound impact in the study of clay - organic interactions than infrared spectroscopy. The combined use of infrared spectroscopy and X-ray diffractometry in studying clay - organic interactions has led to the discovery of significant new data that has increased our understanding of the bonding to and arrangement at the clay surface of the organic species (3).

Infrared Spectroscopy of Aquifer Minerals

In 1964, Farmer and Russell (21) published the results of a detailed study of the infrared spectra of layer silicates. This study, amongst others, provided the much needed background information necessary for the interpretation of the infrared spectra of soil clays. In 1965, Swoboda and Kunze (22), driven by the then increasing concern about the adsorption of insecticides, herbicides, detergents, and other organic compounds by soils, used infrared spectroscopy to study adsorption of volatile vapours by soils and clays. Continuing their studies of the infrared spectroscopy of clay minerals, Farmer and Russell (23) provided more insight into the interpretation of infrared spectra of clay minerals, recognising at the same time that several aspects of the spectra could not be usefully interpreted at that time. Over the years the use of infrared spectroscopy to study clay-organic interactions has continued (24-28).

Fourier Transform Infrared Spectroscopy

The real revolution in infrared spectroscopy was the advent of Fourier Transform Infrared Spectroscopy. That there is very little work in the study of clay minerals and their adsorption complexes using FTIR methods is very unfortunate, considering the fact that FTIR methods are superior to dispersive methods (29,30). Also, the use of FTIR-PAS for studying clay-organic interactions has not received very much attention, in spite of the distinct and dramatic advantages that this technique offers. These advantages were clearly demonstrated by Rockley and coworkers using a simple comparision of the KBr spectrum and the PAS spectrum of a particular soil sample. This test showed the power of FTIR-PAS for observing surface concentrated species in aquifer materials (31).

A brief description of this novel technique is now in order. Radiation absorbed by a gas, liquid, or solid, is converted totally or partially into kinetic energy. If a gas is held in an enclosed chamber, this results in an increase in temperature and hence an increase in the pressure. Modulating the input radiation in the audio frequency range results in the modulation of the pressure fluctuations at the same frequency and this can be detected with a microphone. This is the essence of the photoacoustic effect, also referred to as the photothermal effect (30). This effect has been applied to the spectroscopic study of solid samples with great success. Most often, the solid sample is held in contact with a nonabsorbing gas in an enclosed cell and is then illuminated with the modulated beam from an interferometer. When the sample absorbs some part of the incident radiation, a modulated temperature fluctuation is generated at the same frequency as that of the incident radiation. This causes the layer of gas next to the surface of the solid sample to be heated and an acoustic pressure wave is generated as a result. This acoustic pressure wave can now be detected by means of a suitable microphone (30).

The first use of this method to obtain useful spectroscopic information was reported by Rockley in 1979 (32-34). Applying the two methods - FTIR by KBr pellet transmission and FTIR-PAS - to study clay materials, Rockley et al. (35) observed that FTIR-PAS gives complementary information to that obtained from pellet transmission measurements. The success of FTIR-PAS in many applications provided the basis for the application of this method to obtain spectroscopic information from low TOC soils that could be useful in the prediction of properties of the soils that are linked to pollutant retention by that soil. As has been stated above, the experimental results were encouraging and warranted further investigation.

The tool that led to the development of such a relationship between spectral data and pollutant retention by a soil was factor analysis. Using factor analysis to analyse the FTIR pellet transmission data for ten low TOC soil samples provided by the Robert S. Kerr Environmental Research Laboratory (RSKERL) in Ada, Oklahoma, Rockley et al. (36) were able to show that there was indeed some useful relationship that could be derived from spectroscopic studies of aquifer samples, as was reported earlier. They proposed the further analysis of such samples using Attenuated Total Reflectance (ATR) spectroscopy.

The proposed use of this method finds its basis in the fact that FTIR pellet transmission spectroscopy and FTIR-PAS spectroscopy had limitations which prevented their extensive use. In conventional KBr pellet transmission spectroscopy, the need for fairly elaborate sample preparation and the concomitant subtle alteration in the physical and possibly the chemical composition of the sample poses a serious problem (37). This problem is overcome to a certain extent by the use of FTIR-PAS, which has been proven to have some critical advantages over other infrared analytical techniques in the significant reduction in sample preparation of solid and surface adsorbed species (32-34). This method too has its disadvantages. Reduced sensitivity and certain experimental difficulties associated with sample heating, limitations on particle size, and scattering of the incident radiation, limit the use of FTIR-PAS in the analysis of aquifer materials (35). It is therefore clear that a better method of infrared analysis is necessary. ATR offers several advantages in the studying of aquifer materials by infrared spectroscopy.

ATR Analysis of Aquifer Minerals

ATR is a surface reflectance measurement in which the sample material - solid, liquid, or thin film - is brought close to the reflecting surface, where it interacts with the incident wave, and a spectrum is thus obtained (38). This method is based on the phenomenon of internal total reflection on the interface when the radiation beam comes from the phase of higher refractive index (n1) at an angle of incidence (Q) greater than the critical angle (Fig. 1), and penetrates the phase of smaller refractive index (n2) to a certain depth (dp). It is then possible to obtain a spectrum for the phase with lower refractive index, by obtaining the reflection coefficient, $R=I_R/I_O$ (where I_R is the intensity of the reflected beam andI_O is the intensity of the incident beam) as a function of the wavelength of the incident radiation.

The extent to which the beam penetrates the phase with the lower refractive index (the sample) depends on the optical properties of the system and the wavelength of the incident radiation and the angle of incidence.

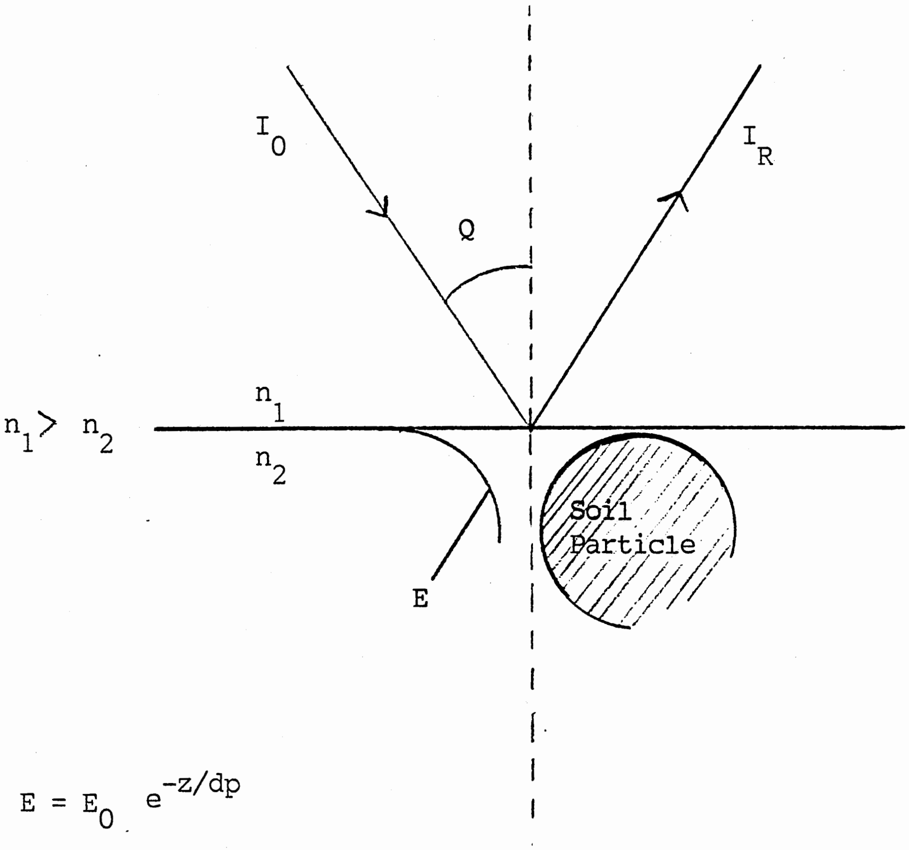


Figure 1 . Schematic diagram of internal total reflection
E_0 = amplitude of electromagnetic wave at interface,
z = distance from the interface in the less dense phase,
dp = depth of penetration, n_1 and n_2 are the refractive indices.
(38)

The same parameters control the intensity of the bands in the resulting reflection spectrum. Since the amplitude of the electromagnetic wave (E) penetrating the sample is inversely related to the distance from the interface, the reflected beam is chiefly composed of information about the surface layers of the sample. Since the optical contact of the sample with the reflecting surface is the only requirement, sample preparation is dramatically reduced. Yet another advantage that this method offers is reduced scattering of the radiation interacting with the sample (39).

Thus, if new and useful measurements are to be made, full use of the latest and best methods of studying aquifer minerals must be made. There remains little doubt that unless reliable methods to reduce, if not eliminate the serious problem of pollution are found, our ecosystem may suffer irreversible damage. To overcome many of the problems associated with FTIR pellet and PAS studies of clay minerals, it is necessary to investigate new methods of studying aquifer minerals by infrared spectroscopy.

Advantages of ATR

ATR offers more advantages over FTIR pellet transmission spectroscopy and FTIR-PAS than any other method in existence today. That this is a very sound method is substantiated by the excellent experimental results obtained so far. Figure 2a shows the FTIR KBr pellet transmission spectrum of EPA10. This spectrum was obtained on a Perkin-Elmer 1710 FTIR spectrometer. Figure 2b shows the FTIR ATR spectrum of EPA10 also obtained using the same instrument. The differences between the two are quite dramatic. The ATR spectrum shows improved band resolution and more definition than the pellet spectrum. This is due to the combination of factors which make ATR a superior method for the analysis of aquifer materials.

Factor Analysis

Spectral acquisition represents the first step in the treatment of the problem. Analysis of the spectra obtained to produce meaningful results is the next step. An FTIR spectrum of a typical sample, represents a fairly large data set. There has been considerable development in the field of analysis and interpretation of large sets of data in the last few years. The application of many mathematical-statistical methods, combined with the increasing power of digital computers, has made it possible to gather information about systems which were previously regarded as too complex and difficult to analyze. One such technique that is relevant to this study is factor analysis.

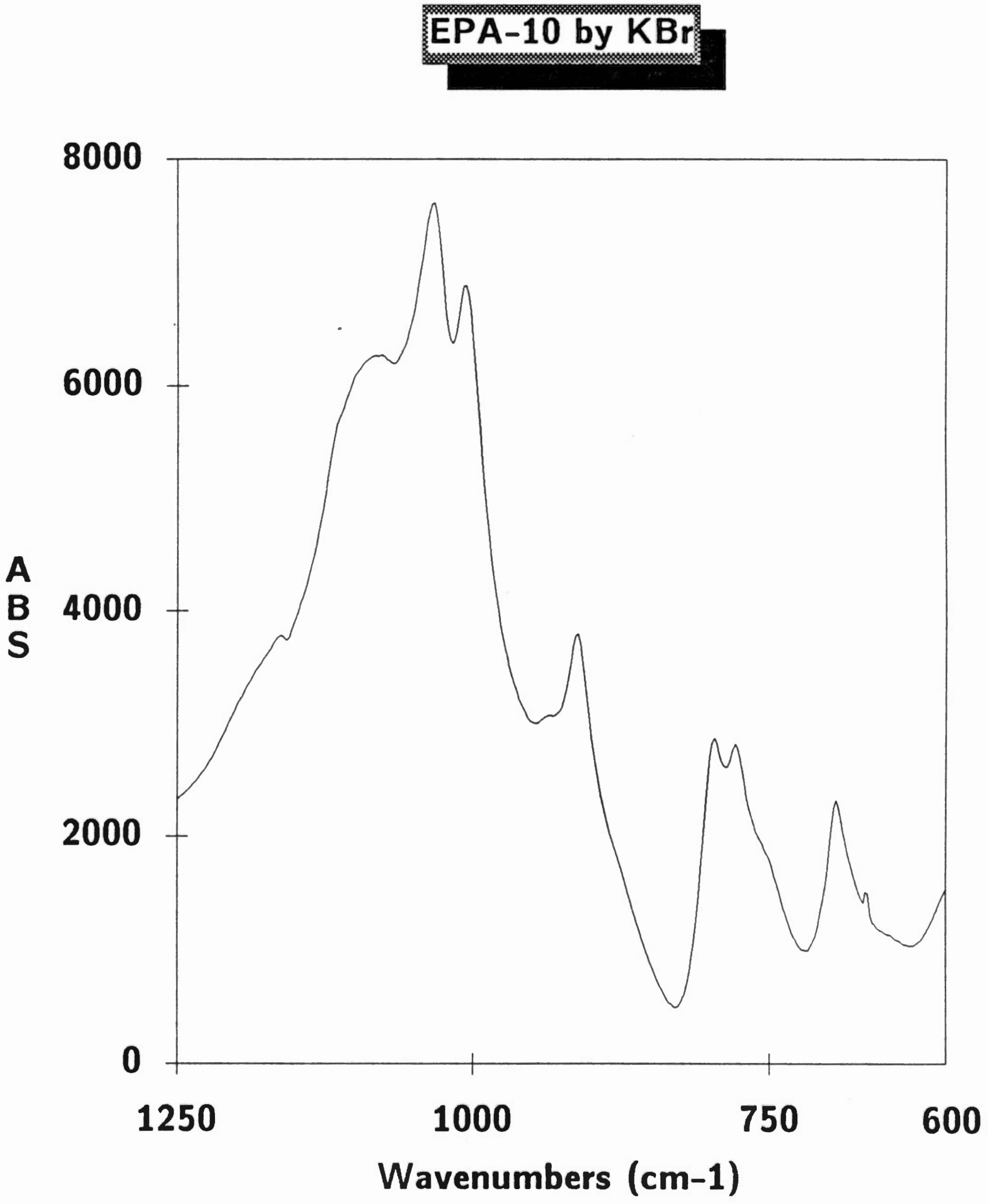


Figure 2a. FTIR Spectrum of EPA10 by KBr pellet transmission

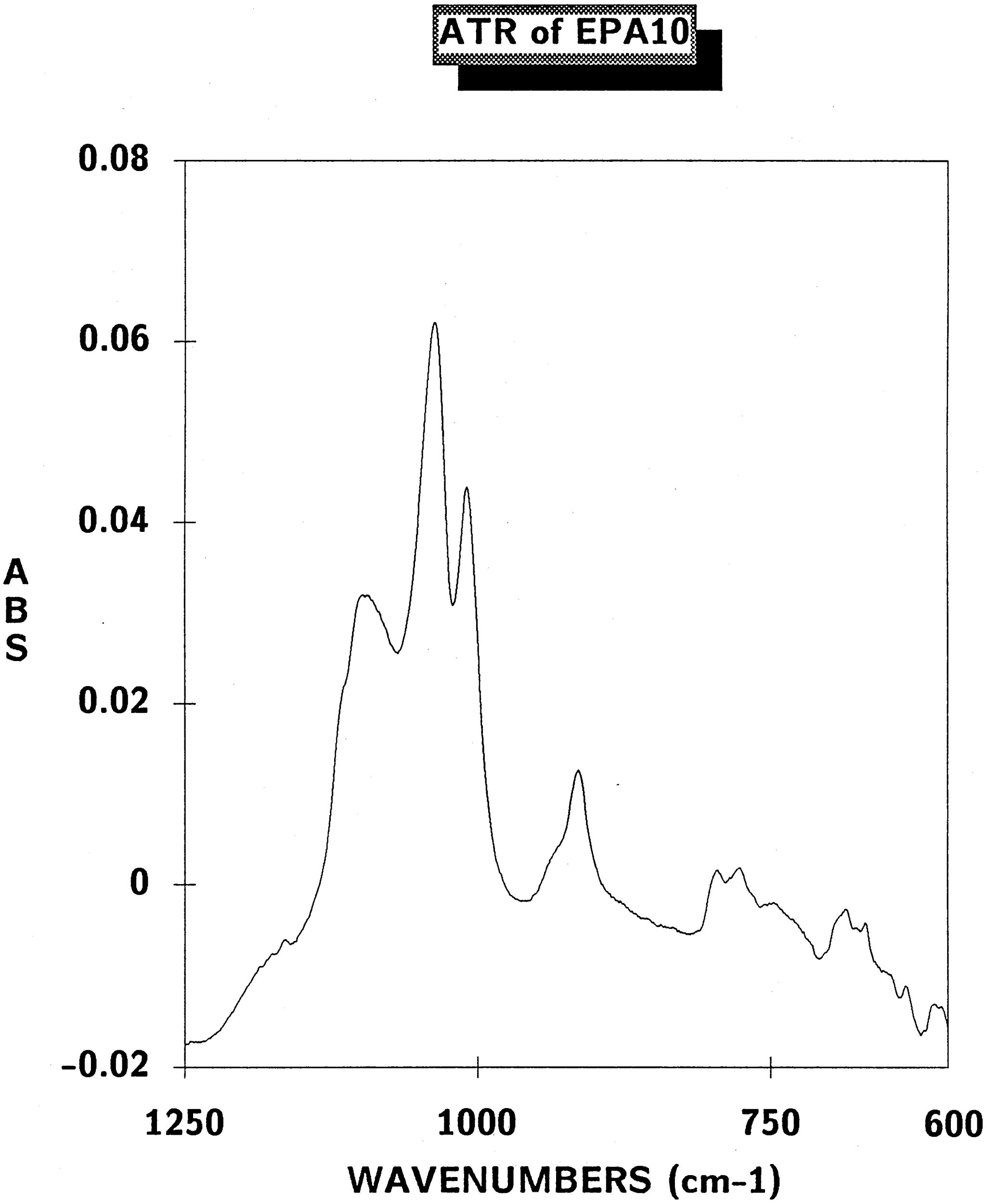


Figure 2b. FTIR Spectrum of EPA10 by ATR

Factor analysis is a very powerful mathematical method for studying matrices of data. This method has been applied successfully to the study of chemical systems for over two decades, and with the increasing use of computers in chemistry, has led to the development of a new subdiscipline of chemistry called chemometrics. Chemometrics has been defined as the science of the application of mathematical and statistical methods for handling, interpreting, and predicting chemical data (40).

Virtues of Factor Analysis

The virtues of factor analysis that make it such a popular tool among chemists might be outlined as follows (40):

1. Factor analysis is capable of handling data of great complexity because it is a method of "multivariate" analysis. This is of particular importance in chemistry, where a multivariate approach is required for the interpretation of most data.

2. The ready availability of computing facilities and the required software make it possible to use factor analysis to analyze large quantities of data.

3. Factor analysis is applicable to the study of many types of problems. "Predictive" analysis can be performed and more insight gained into processes

about which very little is known.

4. It is possible to use factor analysis to simplify data by constructing matrices employing the minimum number of factors to obtain a general pattern describing the data.

5. The abstract factors can be analyzed and manipulated to give physically relevant parameters, which in turn can be used to predict new data.

Main Steps in Factor Analysis

The main steps involved in factor analysis are preparation of data, reproduction of data, target testing, abstract rotation, and prediction. The preparation step involves the mathematical pretreatment of data to be factor-analyzed. In the reproduction step, an abstract solution using the correct number of factors is obtained. The objective of target testing, which is a mathematical transformation step, is to obtain more useful solutions by identifying real factors. Abstract rotation is also a mathematical transformation step that converts the abstract steps into more meaningful abstract factors. In the combination step, complete models of real factors can be derived. The prediction step is the final step and involves the prediction of new data on the basis of the results obtained. A summary of these steps is provided in Table I (40).

In the data preparation step, a data matrix best suited for factor analysis is obtained. Since the success of the entire operation depends to a large extent on the preparation of data, special care must be taken to adhere to the rules for data preparation. In the reproduction step, two procedures are carried out. The first is to obtain the principal factor solution and the second is to determine the correct number of factors using a data reproduction method. Calculation of the abstract solution is based on eigenanalysis. Principal factor analysis (PFA), also called principal component analysis (PCA), is often used to carry out the eigenanalysis. PFA provides an abstract solution containing a set of abstract eigenvectors and a set of abstract eigenvalues, where each principal eigenvector represents an abstract factor. The associated eigenvalue is a measure of the importance of the abstract factor, with a large eigenvalue indicating a major factor and a small eigenvalue indicating a minor factor. The absence of experimental error would yield the exact number of factors n, but since this is never possible, PFA yields c factors, representing each column in the data matrix. Only n of these c eigenvectors are physically meaningful. Factor analysis renders the data matrix (D) into the product

TABLE I

FEATURES OF THE MAIN STEPS IN FACTOR ANALYSIS

Step	Purpose	Procedure	Result
Preparation	Obtain best matrix for FA	Data selection data pretreatment	Complete data matrix in suitable form
Reproduction	Generate abstract model	PFA, stepwise abstract reproduction	Principal factor matrices, number of factors
Transformation			
Target testing	To evaluate test factors individually	Transformation into real factors	Identify real factors
Rotation	To interpret abstract model	Transformation into new abstract matrices	Clustering of data
Combination	Develop models from sets of real factors	Simultaneous transformation into a set of real factors	Key sets of real factors
Prediction	Calculate new data	Free-float missing points, employ key combination set	New target data, new data rows & columns

(Malinowski, Edmund R., and Howery, Darryl G., Factor Analysis in Chemistry, page 21) of an abstract row matrix (R) and an abstract column matrix (C). This is followed by the application of the relevant computer program to calculate the principal factor solution represented by:

$$[D] = [R]_{cPFA} [C]_{cPFA} \tag{1.1}$$

In $[C]_{cPFA}$, the rows represent individual eigenvectors in decreasing order of importance. The first row represents the most important factor and the last row the least important factor, with the whole matrix representing the complete set of data, inclusive of the experimental error.

The determination of the number of factors that are physically important, is the next step. Since the data is composed of real data and experimental error, the number of factors also represent real data and experimental error and can be expressed as two different sets of factors - a primary set of n factors representing real data, and a secondary set of c factors representing experimental error. The elimination of these unwanted secondary factors is called factor compression, and yields

$$[D] = [R]_{nPFA} [C]_{nPFA} \tag{1.2}$$

Equation (1.2) is the general form of the abstract solution. This result forms the basis of all other calculations in factor analysis (40).

Since one of the objectives of factor analysis is to obtain the exact number of factors that completely describe a given set of data, the elimination of experimental error from the data is essential. By its very nature, experimental error cannot be eliminated at the experimental stage. Thus, a mathematical method must be used to achieve this objective. In FTIR-ATR spectroscopy, as in other forms of spectroscopy, the exact nature of the experimental error cannot be determined, at least not in any quantitative manner. It is therefore necessary to account for the experimental error by using methods in which no prior knowledge of experimental error is needed. One such method involves the use the Imbedded Error Function (IE). The imbedded error is a function of the number of factors, the number of rows and columns in the data matrix and the secondary eigenvalues. All this information is always available in factor analysis, irrespective of the nature of the data or the experimental error. Thus by calculating IE as a function of n, as n goes from 1 to c, the number of true factors may be obtained. The variation of the IE with n is usually as follows:

IE decreases with the use of increasing number of primary factors, and starts increasing once secondary factors are used.
The factor at which IE reaches a minimum is usually indicative of the number of true factors representing the given set of data. However, for real data, such a clear pattern is rarely observed because of the fact that nonuniformity of the error is highlighted by the principal component feature of factor analysis. Also, the presence of non-random error, systematic error, or sporadic error will affect the behaviour of the IE function (40).

The Real Error (RE), Extracted Error (XE), and the Imbedded Error (IE) are the three types of error that may exist in a given set of data. These three theoretical errors are related in a Pythagorean way as follows:

$$(RE)^2 = (IE)^2 + (XE)^2 \tag{1.3}$$

where

$$RE = RSD \tag{1.4}$$

$$IE = RSD\ (n/c)^{1/2} \tag{1.5}$$

$$XE = RSD\ ((c-n)/c))^{1/2} \tag{1.6}$$

and

RSD = Residual Standard Deviation

The imbedded error arises because of the fact that in the factor analytical reproduction process, only a fraction of the error from the data is incorporated, and since this error becomes part of the factors, it cannot be removed by repeated factor analysis. Extracted error is the error that is extracted from the data by dropping the secondary eigenvectors (c) from the analysis. While the real error (RE) is indicative of the difference between the pure data and the raw experimental data, the imbedded error (IE) is indicative of the difference between the pure data and the data reproduced from factor analysis. From equation (1.5), it can be seen that

$$\text{for } n < c, \quad IE < RE \tag{1.7}$$

Hence the error between the data reproduced by factor analysis and the pure data is less than the original error between the raw data and the pure data. This means that even without the knowledge of the real factors, data can always be improved merely by employing more than n (primary factors) columns in the data matrix (40). The use and significance of one more error function, the factor indicator function is explained in Chapter III of this thesis.

Transformation of the principal factors into physically useful parameters represents the most significant use of factor analysis. Transforming the abstract factors into more recognizable physical factors involves the use of a suitable transformation matrix [T] and its inverse $[T]^{-1}$. The transformation is carried out as follows:

$$[D] = [R]_{PFA} [C]_{PFA} \qquad (1.8)$$

$$= \{[R]_{PFA} [T]\}\{[T]^{-1} [C]_{PFA}\} \qquad (1.9)$$

$$= [R]_{transformed} [C]_{transformed} \qquad (1.10)$$

If the transformation is successful, the transformed matrices will represent factors that are physically more meaningful. The transformation process can be carried out in one of two ways. Target transformation, which handles factors one at a time, or abstract rotation in which one of many mathematical techniques may be used to transform PFA abstract matrices into more significant abstract factors. Combination involves the reproduction of the data from real factors as opposed to abstract factors. Finally, the prediction step enables one to predict new data by means of a procedure called free floating. A detailed explanation of these methods may be found in Chapter 2 (pp 10-22) of the monograph by Malinowski and Howery. (40)

Example of Factor Analysis in Chemistry

An example of the application of factor analysis to a chemical problem is now in order. Consider a hypothetical data matrix [A], representing the absorbances of four different mixtures of the same absorbing components, at five different wavenumbers:

$$[A] = \begin{array}{c|cccc|}
 & & \text{mixture} & & \\
\text{wavenumber} & 1 & 2 & 3 & 4 \\ \hline
1 & 0.371 & 0.713 & 0.219 & 0.186 \\
2 & 0.271 & 0.515 & 0.202 & 0.174 \\
3 & 0.229 & 0.424 & 0.241 & 0.271 \\
4 & 0.349 & 0.641 & 0.409 & 0.428 \\
5 & 0.182 & 0.226 & 0.229 & 0.265 \\ \hline
\end{array} \qquad (1.11)$$

(From page 6, chapter 1 of the monograph by Malinowski and Howery. (40))

This information is typical of many types of spectroscopic measurements commonly used in chemistry. The problem represented here is twofold - to find the number of components present and to determine their concentrations. According to the general solution of factor analysis, we obtain

$$A_{ik} = \sum_{j=1}^{n} w_{ij} m_{jk} \qquad (1.12)$$

where

w_{ij} = jth abstract row cofactor associated with the ith wavenumber

m_{jk} = jth abstract column cofactor associated with the kth mixture

and

A_{ik} = absorbance data point.

The inclusion of n factors in the sum accounts for absorbances within experimental error. Then, from equation (1.12), we obtain the factor analytical solution

$$[A] = [W]_{\text{abstract}} [M]_{\text{abstract}} \tag{1.13}$$

where

$[W]_{\text{abstract}}$ = wavenumber-factor matrix

and

$[M]_{\text{abstract}}$ = mixture-factor matrix.

This abstract solution gives an indication of the number of factors responsible for the absorbance data in (1.11). By using a suitable transformation matrix, the abstract factor matrix may be converted into a matrix of real and physical solution given by

$$[A] = [W]_{\text{real}} [M]_{\text{real}} \tag{1.14}$$

The transformation process is the most difficult step in this analysis. The application of theoretical considerations to the transformation process increases the probability of the transformation process being successful. This leads us to the fact that Beer's law can be used to represent absorbance data for multicomponent systems, and hence enable us to interpret the factors chemically. On the basis of Beer's law, we obtain

$$A_{ik} = \sum_{j=1}^{n} \epsilon_{ij} c_{jk} \tag{1.15}$$

where

ϵ_{ij} = molar absorptivity per unit path length of component j at wavelength i

and

c_{jk} = molar concentration of component j in the kth mixture.

Since this is a linear sum of products similar to (1.12), it follows that data that follow Beer's law must have factor analytical solutions as well. The complete solution of this problem will involve the transformation of this (1.15) abstract solution into the real solution. If this is carried out correctly,

$$[A] = [E]_{real} [C]_{real} \tag{1.16}$$

is obtained as the real solution. $[E]_{real}$ is the molar absorptivity matrix in which each column corresponds to the absorbance of one of the pure components at the five different wavelengths, providing the spectrum of the pure component. $[C]_{real}$ is the molar concentration matrix in which each row corresponds to the concentration of one of the n components in each of the four mixtures.

Thus, by determining the number of factors it is possible to determine the number of absorbing components in the mixtures. The transformation process results in the chemical identification of each component through its spectrum obtained from the molar absorptivity matrix, $[E]_{real}$, while the molar concentration matrix $[C]_{real}$, provides the concentration of each component in the four mixtures. Thus, the factor analysis of the absorbance data leads to the complete solution of the problem (40). The solution for the data presented produces a factor size of 3, indicating that there were 3 absorbing components in each mixture. This result was found to be correct, since the data represented in (1.11) was obtained from a 3 component system (40).

The above example, although a simple one, adequately illustrates the power of factor analysis in solving many types of problems in chemistry. Many difficult and complex chemical problems can be modelled after this example , making it possible to attempt solutions that were previously unattainable.

The FTIR spectra of aquifer materials can be related to many of the significant practical properties of that material, using factor analysis. Fredricks et al. (41) applied factor analysis to develop a method for the characterization of coal, bauxite, manganese dioxide ore, and diesel fuel and concluded that such a method would be applicable to any material that exhibits an infrared spectrum. The same authors provide an excellent description of the application of factor analysis to interpret FTIR spectra in another publication (42).

Presentation of the Problem

The problem of pollutant sorption by aquifer minerals is far more complex than many of the problems that have been studied so far, mainly because of the serious lack of background information and the complexity of the clay system. This makes it a likely candidate for factor analysis, since factor analysis can handle such data quite reliably. The use of FTIR-ATR spectroscopy to study aquifer mineral - organic pollutant interaction, has a better chance of providing data of superior quality that can be analyzed using factor analysis, eventually leading to the development of a significant and sound relationship between aquifer minerals and the sorption and migration of organic pollutants. It is hoped that the development of such a relationship will verify and strengthen the findings of previous studies involving the use of KBr pellet transmission spectroscopy and FTIR-PAS spectroscopy, which have indicated that there does exist a set of high correlation factors, which predict - albeit semi-quantitatively - the relationship between aquifer minerals and the sorption of organic pollutants by them (29).

CHAPTER II

EXPERIMENTAL PROCEDURES

Materials

Ten aquifer minerals were obtained from the Robert S. Kerr Environmental Research Laboratory (RSKERL) in Ada, Oklahoma. The samples were obtained by the personnel of RSKERL, using a procedure outlined in the paper by Banerjee et al. (17). These ten samples are labelled as EPA-1 to EPA-10 and are known to the EPA lab as J-10 for EPA-4, N-6 for EPA-6, C-1 for EPA-8, and B-1 for EPA-9. A description of the physical appearance and texture of each of the ten samples is provided in Table II. Measured partition coefficients (Kp) for four different solvents, Benzene (BZ), Trichloroethene (TCE), Tetrachloroethene (PCE), and 1,2-Dichlorobenzene (DCB), determined by RSKERL on four of the ten aquifer materials, were provided by RSKERL and are listed in Table III. The partition coefficient Kp, is a value determined from the extraction of the solid phase (sorbent) in (L/kg). This is based on the total mass of the sorbate (pollutant) recovered from the sorbent (aquifer mineral) by extraction. An Agate

TABLE II

PHYSICAL DESCRIPTION OF AQUIFER MINERALS EPA1 - EPA10

Sample #	Color	Texture
EPA1	Dark Brown	Fine
EPA2	Light Brown	Coarse
EPA3	Dark Brown	Fine
EPA4	Med. Brown	Fine
EPA5	Med. Brown	Coarse
EPA6	Yellow-Brown	Medium
EPA7	Yellow-Brown	Medium
EPA8	Light Brown	Sandy
EPA9	White-Brown	Sandy
EPA10	Pink	Fine

TABLE III

MEASURED PARTITION COEFFICIENTS FOR SOLVENTS ON AQUIFER SOLIDS (CI = 90% CONFIDENCE INTERVAL)

Sorbent/Solvent		BZ	TCE	PCE	DCB
Eq Kow		2.1	2.3	2.9	3.4
EPA9	Kp	0.038	0.088	0.35	0.34
	CI	.030-.045	.076-.098	.33-.37	.31-.37
EPA8	Kp	0.026	0.032	0.19	0.082
	CI	.021-.031	.029-.036	.17-.21	.072-.091
EPA6	Kp	0.035	0.076	0.18	0.23
	CI	.031-.039	.069-.083	.17-.19	.22-.24
EPA4	Kp	0.12	0.16	0.48	1.04
	CI	.11-.13	.15-.18	.45-.50	1.01-1.08

Kp = aquifer mineral / pollutant partition coefficient

Kow = octanol / water partition coefficient mortar and pestle was used for grinding the soil samples. The unground particle size of the ten samples varied from approximately 220 microns for the fine samples to 850 microns for the coarse samples. The particle size of the samples after grinding was found to be approximately 30 microns. This measurement was carried out on a Cambridge Instruments Stereoscan - 90B Scanning Electron Microscope. The distilled, deionized water used in the sample slurry preparation for examination by ATR was obtained each day from facilities in the department of chemistry, OSU.

Sample Preparation

Approximately 50 mg of each sample was used for the acquisition of each spectrum. The soil was weighed out using a clean spatula and Lilly brand powder paper (glassine) on a Sartorius electronic balance. The soil sample was then transferred to a clean agate mortar, and after the addition of 7 - 10 drops of distilled deionized water (approx. 0.8 ml) was ground to a fine paste. The grinding time varied with the coarseness of the sample, and in most cases was between 5 and 7 minutes, and the final particle size obtained was approximately 30 microns. No further sample preparation was required.

X-Ray Analysis of the Soils

X-Ray analysis of the soils to determine the mineral content (XRD analysis) was performed by Minerology Inc., a commercial lab based in Tulsa, Oklahoma. This analysis was done in duplicate for samples EPA-6 to EPA-10. The XRD data for all ten samples is listed in Table IV. It must be noted that the X-Ray data listed in Table IV are not error free. Lack of precision, or reproducibility of results, whose quantitative nature is questionable, is a serious problem associated with XRD analysis. These data must be subject to careful scrutiny, before any conclusions based on it are made. Although X-Ray analysis is the most popular tool for qualitative analysis, its use for quantitative analysis is subject to problems associated with preferred orientation, extinction and micro-absorption, resulting in relatively inaccurate or qualitative results at best (43,44).

Infrared Spectroscopy

The instrument used to measure the ATR - mid IR spectra was a PERKIN-ELMER 1710 FTIR Spectrometer. This spectrometer is equipped with a temperature stabilised, coated FR-DTGS (Fast Recovery Deuterated Tri Glycine Sulphate) detector with moisture resistant CsI window. The source is a temperature stabilised

TABLE IV

DATA FROM XRD ANALYSIS OF EPA SAMPLES EPA1 - EPA10

MINERAL	EPA1	EPA2	EPA3	EPA4	EPA5
VERMICULITE	0	0	0	0	0
CHLORITE	1	0	trc	trc	0
FELDSPAR	7	6	0	2	3
DOLOMITE	trc	0	0	0	0
QUARTZ	88	92	98	94	94
ILLITE	2	1	1	1	2
MTML*	trc	trc	0	1	trc
CALCITE	1	trc	trc	0	0
GYPSUM	0	0	0	0	0
KAOLINITE	1	1	1	2	1

MINERAL	EPA6	EPA7	EPA8	EPA9	EPA10
VERMICULITE	0	0	0	0	0
CHLORITE	0	0	0	0	trc
FELDSPAR	2/4	4/3	1/1	16/28	1/trc
DOLOMITE	0	0/trc	0	1/trc	trc/0
QUARTZ	95/89	92/93	96/97	70/52	85/82
ILLITE	2/2	2/2	2/1	0	trc
MTML*	trc	0	trc	0	0
CALCITE	0/trc	0/trc	0	9/15	trc
GYPSUM	0	0	0	1	0
KAOLINITE	1/2	1/2	1/1	0	11/14
HALLOYSITE	0	0	0	0	3/4
AMPHIBOLE	0/2	1	0	0	2/4

* MTML = MONTMORILLONITE

trc = trace
2/4 = duplicate measurement
precision = 2% on one analysis
4% on the second analysis of same sample

(Data obtained from Minerology Inc., Tulsa, Oklahoma).

ceramic source operating at 1400 K. The abscissa range is from 4400 cm-1 to 400 cm-1, with an accuracy of 0.01 cm-1 using a He-Ne laser reference. The signal to noise is better than 0.1 %T peak to peak, 0.025 %T RMS for a 4 second measurement at 4 cm-1 within the range 2200 cm-1 to 2000 cm-1. The typical scan time is 1 second per scan at a resolution of 4 cm-1. Since a spectrometer measures only a finite part of the interferogram, the output spectrum has unwanted oscillations or "feet" on either side of sharp spectral bands. Apodization, a mathematical operation, reduces these oscillations, but with a slight loss of resolution. The apodization function used was the MEDIUM Norton - Beer apodization, one of 9 types available on the PE-1710. The Jacquinot stop (J-stop) is an aperture placed in the beam path between the source and the interferometer. It reduces beam divergence which could degrade resolution (30). The default J-stop of 2 was used to obtain all the spectra. This allows a maximum resolution of 2 cm-1 over the entire abscissa range. Spectral smoothing was not done. (45) The instrument has a single beam and the sample spectra are obtained from the ratio of spectra with a sample in the beam against background spectra obtained without a sample. The ATR accessory was positioned carefully and adjusted to give the maximum energy reading on the spectrometer. The details of spectral acquisition are explained in the next section. The instrument is controlled by a Perkin-Elmer Series 1700 intelligent controller. The spectra obtained can be stored in one of the three memory areas of the Video Display Unit, (VDU). Conditions under which the spectra are to be obtained are user definable. The conditions under which all the spectra in this study were obtained are as follows:

Abscissa Range ---- 4000 cm-1 to 600 cm-1
Resolution -------- 4 cm-1
Number of Scans --- 256 scans

The time required to obtain one spectrum was 19.2 minutes, being the time taken in the slow scan mode. In the fast scan mode the time required for 256 scans at a resolution of 4 cm-1 is 9.5 minutes. The sample chamber of the spectrometer is maintained under a constant Nitrogen gas purge (liq. N_2 boil off) to reduce spectral interference by water vapor and carbon dioxide.

The attenuated total reflectance accessory used in this study was a Harrick Scientific PLC-11M Single Reflection Prism Cell. This cell has an optical power transmission coefficient of 40% to 60%. This cell is simple in design and is very suitable for recording spectra of aqueous solutions or slurries under ambient conditions. This cell eliminates the need for elaborate sample handling procedures. The entire assembly is mounted in the spectrometer using a standard sample slide plate (see fig. 3). The soil samples, which have been ground into a fine paste, are simply spread on the surface of the prism and then covered with a glass plate. (Approximately 1.5 ml of the slurry is spread over the entire surface of the sample ATR prism). After the acquisition of each spectrum, the prism is rinsed carefully with distilled water and dried off with Kleenex brand facial quality tissue. The prism is made of ZnSe and due to its toxicity, special precautions such as the use of gloves, must be taken in handling the prism. Figure 3 provides an exploded view of the ATR cell (46).

The first step in obtaining a spectrum was to obtain a background spectrum. This was done by using the ATR attachment without any sample. The next step was to obtain the spectrum of pure water by spreading a thin film of pure water on the prism. Following this, each of the ten samples were scanned under the conditions stated earlier. As each spectrum was obtained, it was stored in one of the memory areas of the spectrometer. The spectrum of water was subtracted

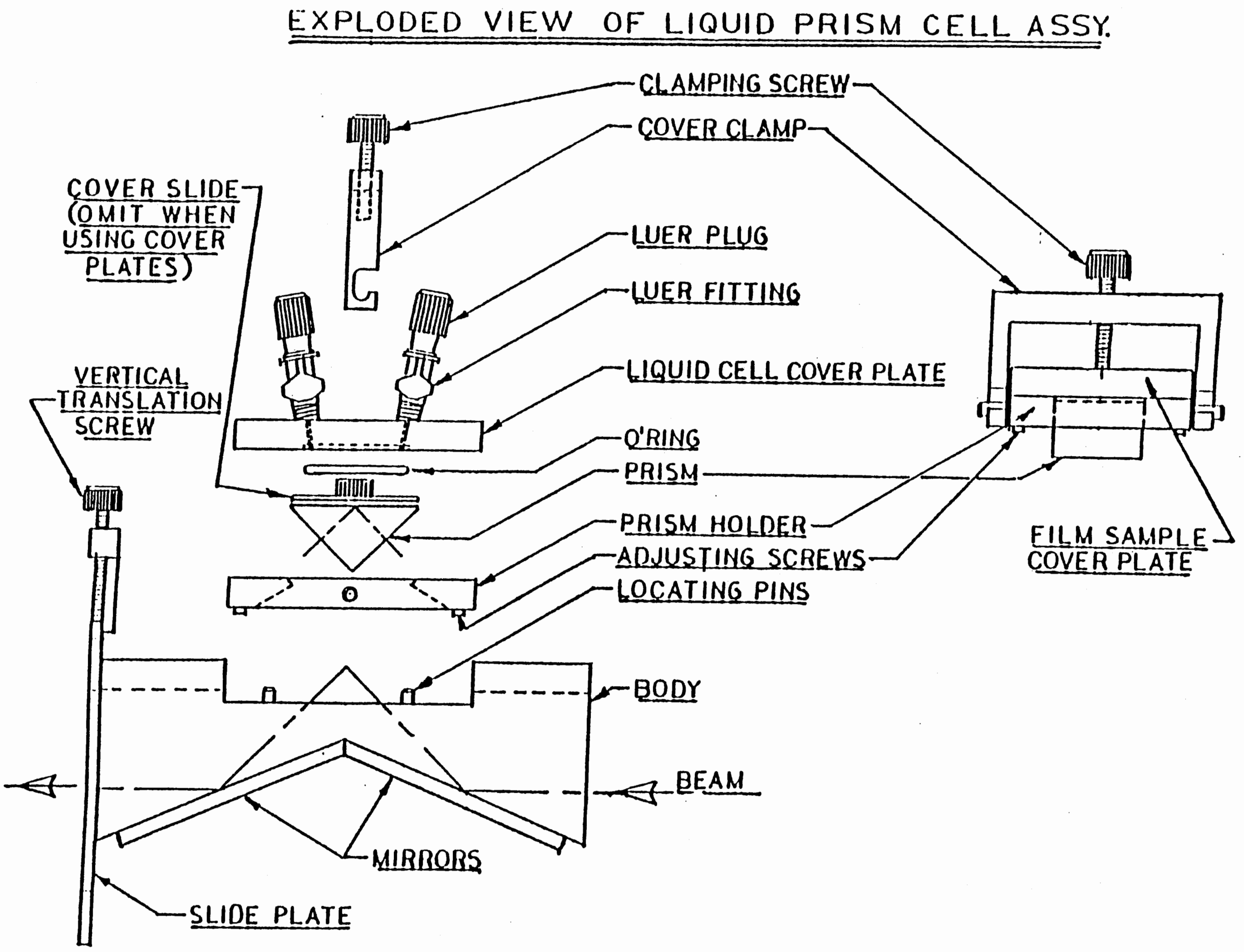


Figure 3. Exploded view of Harrick PLC-11M ATR cell. (from instruction manual for Harrick 11M ATR cell)

from the sample spectrum. This difference spectrum was then transferred to the external computer and stored on disk. Each spectrum was physically checked for accuracy by comparing the data points transferred to the external computer with the data points displayed on the spectrometer. The background spectrum was scanned each day. The spectra of all ten soil samples were obtained in duplicate runs.

Data Acquisition and Transfer

Although the Perkin-Elmer spectrometer can scan and store spectra in one of its three on-board memories, it does not have the capability to store the spectral data on disk, without the acquisition of an external data station. However, the factor analysis program required the spectral data to be digitized and stored on disk. This transfer from spectrometer to computer was accomplished by interfacing the Perkin-Elmer spectrometer to a CORDATA PC 400 personal computer, equipped with a 20 megabyte hard disk and upgraded to IBM AT performance. One of the key elements in establishing the computer-spectrometer interface was the use of a Smart Cable model SC821PLUS Smart Cablemaker, manufactured by IQ Technologies, Inc., which permits the user to establish the correct communications parameters between the computer and the

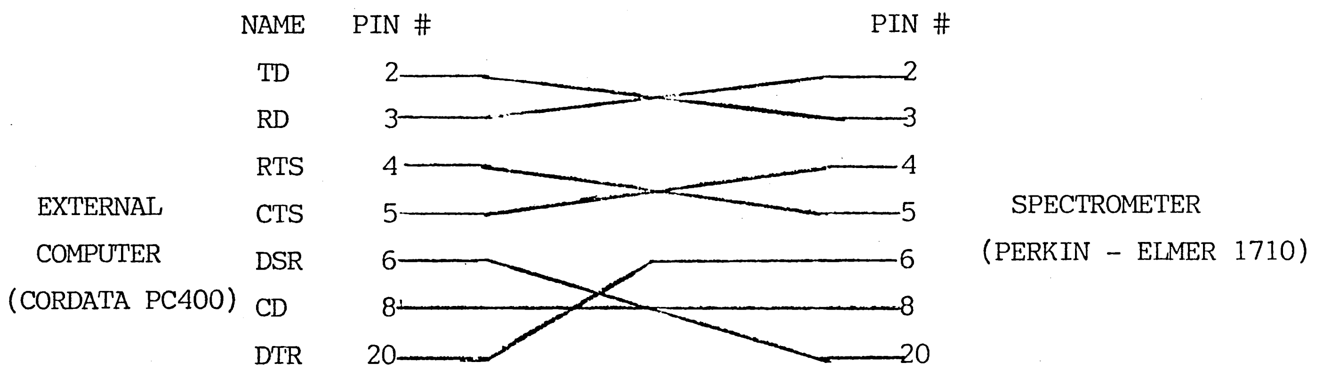


Figure 4. Schematic (cable) diagram of RS232 serial interface between CORDATA PC 400 and PERKIN - ELMER 1710 spectrometer

spectrometer. Fig. 4 shows the cable diagram (47). Since no suitable data acquisition software was available, a program was written in the ‘C’ programming language. This program was developed using Microsoft’s ‘C’ programming language software version 4.0 (48-53). The program transfers control of selected commands of the spectrometer to the external computer and permits the user to transfer spectra to the computer and store it on disk. Further, the program converts the incoming spectral data into a form acceptable by the factor analysis program. The ‘C’ program was subjected to extensive testing to verify the authenticity of the data being transferred. This testing included ‘worst-case’ testing, in which spectra with maximum possible absorptions were transferred and tested for accuracy. (A complete listing of the data transfer program can be found in appendix A). Before the spectra could be subjected to factor analysis, the data files were formatted to meet the requirements of the factor analysis program using the EDIX text editor, available from Emerging Technologies Inc..

Factor Analysis

Factor analysis of the data was performed by using a computer program written in ‘C’. This program was originally developed by Rockley (20) on the basis of algorithms outlined in Malinowski and Howery (40). This program required further modification to accept spectral data from the Perkin-Elmer spectrometer. (A complete listing of both the original code and the modified code can be found in appendix A). The modifications to the original program included changes in the size of the spectral data arrays, the range of analysis and the number of points analysed. The factor analysis program could handle ten spectra at the same time. (The program can be modified to handle more spectra). This program was executed on a Cordata PC 400 personal computer with a 20 megabyte hard disk, upgraded to IBM AT performance. The factor analysis returns the eigenvalues and the loadings on the abstract mathematical factors into which a set of ten spectra (representing the ten aquifer minerals) have been compressed. Factor analysis was carried out for various spectral ranges. The range between 1400 cm-1 and 600 cm-1 contained most of the spectral information. It was decided to limit the factor analysis to the same spectral range. The data from the factor analysis can be found in appendix B.

This data was then transferred to an IBM PS2/80 personal computer for graphical presentation using Microsoft Excel, version 2.0.

CHAPTER III

RESULTS AND DISCUSSION

It is reasonable to assume that some simple property of the aquifer minerals may lead to the development of a relationship between pollutant sorption and Kp values of typical nonpolar pollutants. The experiments performed as part of this study show that a relatively simple and well established spectroscopic technique might be used to characterize aquifer minerals to better understand the transport of nonpolar pollutants through low TOC soils.

Mineral analysis on these samples was performed by an external lab using XRD, to obtain a quantitative estimate of minerals in the ten samples. The results from this XRD analysis are shown in Table IV. Results were obtained in duplicate for samples EPA6 to EPA10, and where available, these values are the second set of numbers. As was stated earlier the procedure used for this quantitation - XRD, is subject to considerable error and hence these values provide us only with approximate indications of content. The data obtained show that all ten aquifer minerals contain five minerals as the principle components.

These are : feldspar, quartz, illite, calcite and kaolinite.

Measured partition coefficients (Kp) for the four different solvents used were determined by RSKERL on four of the ten aquifer minerals being studied and are shown in Table II.

Abstract factors

FTIR-ATR spectra of the ten aquifer minerals provided by RSKERL were measured by the procedure outlined earlier (see appendix C for spectra). The quantitative prediction of aquifer retention properties first involved the transformation of the matrix of spectra by factor analysis procedures to obtain four abstract factors for FTIR-ATR spectra. These four abstract factors were found to be sufficient to describe all ten aquifer minerals. That there are only four significant factors is supported by the fact that the factor indicator function for the factor analysis of the ten spectra between 1400 and 600 cm-1 reaches a minimum at the fourth factor. The factor indicator function specified by Malinowski (40) is an empirical function which appears to be more reliable than the imbedded error (IE) function in its ability to define the correct number of abstract factors describing the system. The indicator function reaches a minimum when the correct number of factors are employed. This minimum is more pronounced than the minimum in the IE function and more importantly, often occurs in situations in which the IE function exhibits no minimum. It must be noted however that the indicator function is not fully understood at this time and that it must be used cautiously. The results of this analysis between 1400 and 600 cm-1 are shown in Table V. Although the XRD analysis would seem to indicate the presence of five principal components, the fact that the factor analysis indicates only four principal components suggests that only four of those five components are truly significant. While the four abstract factors from the factor analysis do not directly correspond to the components observed by XRD analysis, it is fairly certain that there are four components which would emerge after suitable rotation procedures such as automated spectral isolation (ASI) described by Lin and Liu (54) and Lin and Lin (55). They used automated spectral isolation (ASI) to successfully isolate component spectra from the spectra of mixtures. One of the advantages of ASI is that it does not require regions of spectral purity. In ASI, the first step is to normalize each digitized spectrum such that the maximum absorption is 1.000. Principal factor analysis (PFA) is then carried out to define the

TABLE V

ABSTRACT FACTOR ANALYTICAL REDUCTION OF A SET OF TEN FTIR-ATR SPECTRA

FREQUENCY RANGE : 1400 cm-1 to 600 cm-1

Eigenvalue	RE	XE	IE	IND
36.81734	0.01497	0.01497	0.00000	0.00015
1.24248	0.00875	0.00830	0.00277	0.00011
0.38398	0.00511	0.00457	0.00228	0.00008
0.08254	0.00388	0.00325	0.00212	0.00008
0.03728	0.00313	0.00242	0.00198	0.00009
0.02639	0.00227	0.00161	0.00161	0.00009
0.01200	0.00164	0.00104	0.00127	0.00010
0.00495	0.00124	0.00068	0.00103	0.00014
0.00215	0.00097	0.00044	0.00087	0.00024
0.00152	0.00000	0.00000	0.00000	0.00000

Where :

RE = real error
XE = extracted error
IE = imbedded error
IND = factor indicator function

primary eigenvectors and, hence the number of components. Finally, to define the spectral axes for the pure components, "prototype spectra" are target tested and a risk function is used to judge the predicted vectors. The smaller the risk function, the closer the predicted spectrum is expected to match the spectrum of the pure component. A detailed analysis of the exact procedure used in ASI is outlined in chapter 7 of the book on factor analysis by Malinowski and Howery (40).

If these four factors that correspond to the abstract factors can be reliably determined, it is possible to obtain the spectra of those components which are chiefly responsible for pollutant sorption by the soils tested in this set of samples. Once these spectra are obtained it will be possible to better understand the mechanism of pollutant sorption on low TOC soils.

Factor Analytical Compression

Factor analytical compression was done to obtain the loadings of the aquifer minerals (EPA1 - EPA10) on each of the abstract factors. In factor analysis the raw data is not used for analysis. Raw data is first converted into a covariance or correlation matrix. By the application of standard mathematical techniques, the covariance or correlation matrix is decomposed into a set of abstract factors. These abstract factors, when multiplied, reproduce the original data. The reason these factors are called abstract is because, although they have mathematical meaning, they do not have any physical or chemical meanings in their present form. These abstract factors are then converted into physically meaningful parameters by target transformation.

Target Transformation

Target transformation requires an intimate knowledge of the theoretical aspects that form the basis of infrared spectroscopy. This process is preceded by factor compression, in which unwanted factors are dropped. The reproduction of the original data, in this case an FTIR spectrum, from the minimum number of eigenvectors is a very challenging process. Not all the eigenvectors are required. It is seen that the magnitude of the eigenvalue is a measure of the importance of the corresponding eigenvector. The higher the magnitude of the eigenvalue, the more important is the corresponding eigenvector. It stands to reason therefore, that since the smallest eigenvectors are the least important, they may be dropped from the analysis. It has been observed that the inclusion of these insignificant eigenvectors often leads to the re-introduction of experimental error. Hence, the retention of unnecessary eigenvectors is not profitable. This process of dropping the unwanted eigenvectors is called factor compression (40).

At this point it was necessary to eliminate six of the ten aquifer minerals from the study due to the unavailability of Kp values. The four samples retained were EPA4, EPA6, EPA8, and EPA9. It was further necessary to drop EPA4 from the study since it was learnt that sample number EPA4 (known to RSKERL as J-10) was a high TOC soil. Thus the final analysis was based on the use of three samples only - EPA6, EPA8, and EPA9.

Normalization of Factors

The abstract factors from the factor analysis for these three soils were then normalized. The normalization was done as follows:

1. Obtain the sum of the squares of the four factors, for each of the three soils.
2. Take the square root of this sum.
3. Divide each factor of each sample by the corresponding sum to obtain the normalized factor.

If C1, C2, C3, and C4 are the four factors for a particular sample, the normalization process can be expressed as follows:

step 1:

$$SUM = \{(C1)^{2} + (C2)^{2} + (C3)^{2} + (C4)^{2}\} \qquad (3.1)$$

step 2:

$$SQSUM = SQRT\ (SUM) \qquad (3.2)$$

step 3:

divide each factor by SQSUM to obtain the normalized factor. This is given by

$$NORMC1 = (C1/SQSUM) \qquad (3.3)$$

$$NORMC2 = (C2/SQSUM) \qquad (3.4)$$

$$NORMC3 = (C3/SQSUM) \qquad (3.5)$$

$$NORMC4 = (C4/SQSUM) \qquad (3.6)$$

where NORM# is the normalized factor, for a particular aquifer mineral sample. This process is repeated for the three EPA samples analyzed.

Relationship between Kp and Abstract Factors

After normalization it was found that at least for the four sets of Kp values for the range of solvents obtained by RSKERL so far, there was a good linear fit which described the relationship between the Kp values for a given solvent and the corresponding loadings on abstract factors 2 and 4 (C2 and C4) for the three soils analysed, as shown in figures 5(a-d) and 6(a-d), respectively. The one exception was in the case of

PCE, for which the abstract factor C1 gave the best linear fit, (figure 7c) followed by C4. Abstract factors C1 and C3 also correlate fairly well (figures 7(a-d) and 8(a-d), respectively). The slopes and the ordinate intercepts of these linear fits were then plotted against the Kow values for the solvents used. This resulted in a good correlation with a linear fit, for the abstract factors 2 and 4 (C2 and C4) as shown in figure 9(a,b), respectively. These results, while limited in scope due to the small number of Kp values provided by RSKERL, suggest a useful way to define and understand pollutant - soil interactions. For a given soil - pollutant system, the Kow (octanol-water partition coefficient) value of the pollutant is obtained from literature. From the plot of Kp vs loading on the physically relevant factors, in this case C2 and C4, determine the offset and slope of the linear fit, for that particular pollutant. Next, measure the FTIR-ATR spectrum of the soil and factor compress to obtain the loadings on factors C2 and C4 for that soil. Finally, from the solvent Kp versus C2 or C4 lines for the general case (as defined by this work) and the value of C2 or C4 obtained from the spectral - FA measurement, calculate Kp for the solvent of interest on that particular soil.

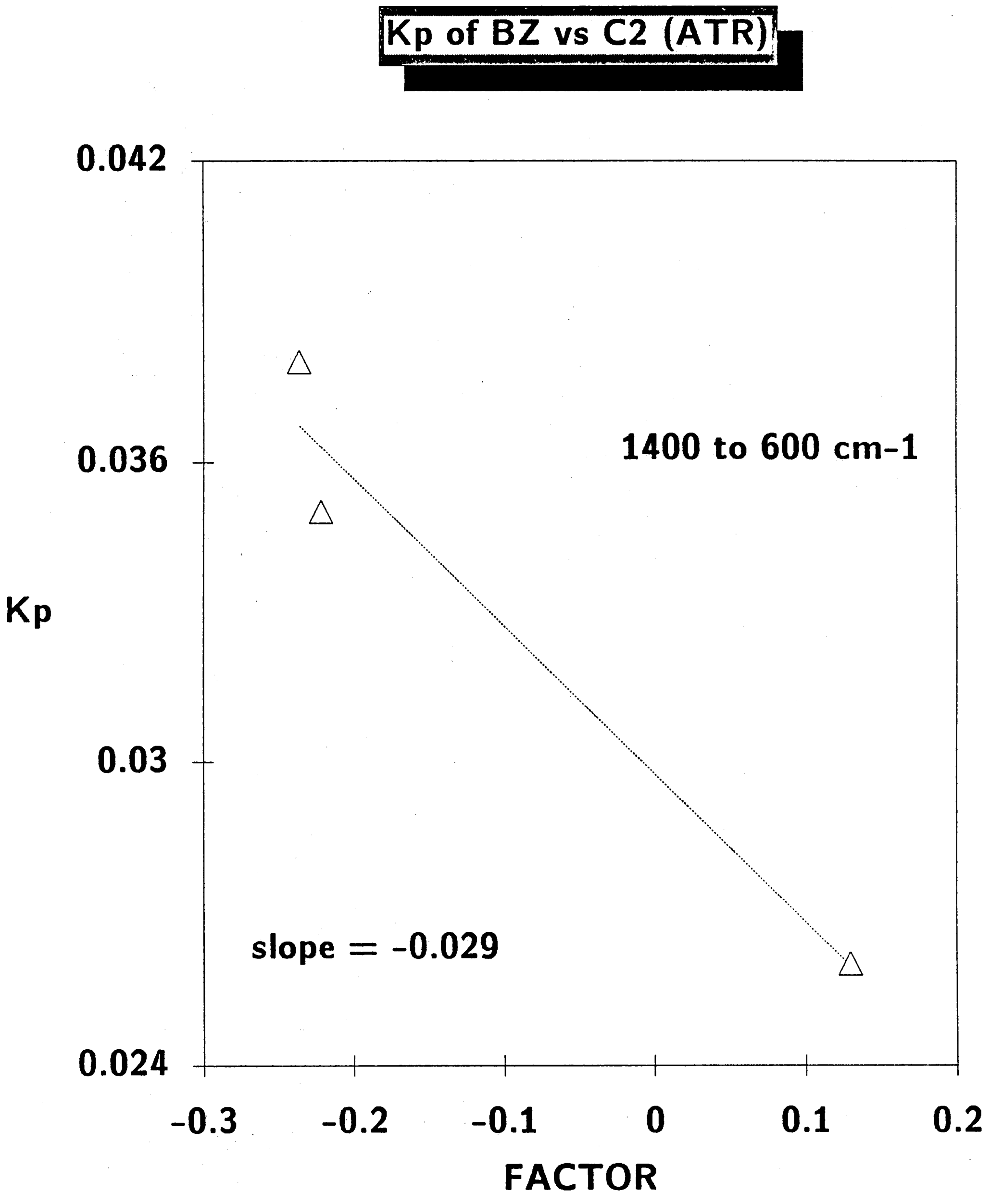


Figure 5a. Kp of BZ vs loadings on abstract factor C2 for EPA6, EPA8, and EPA9

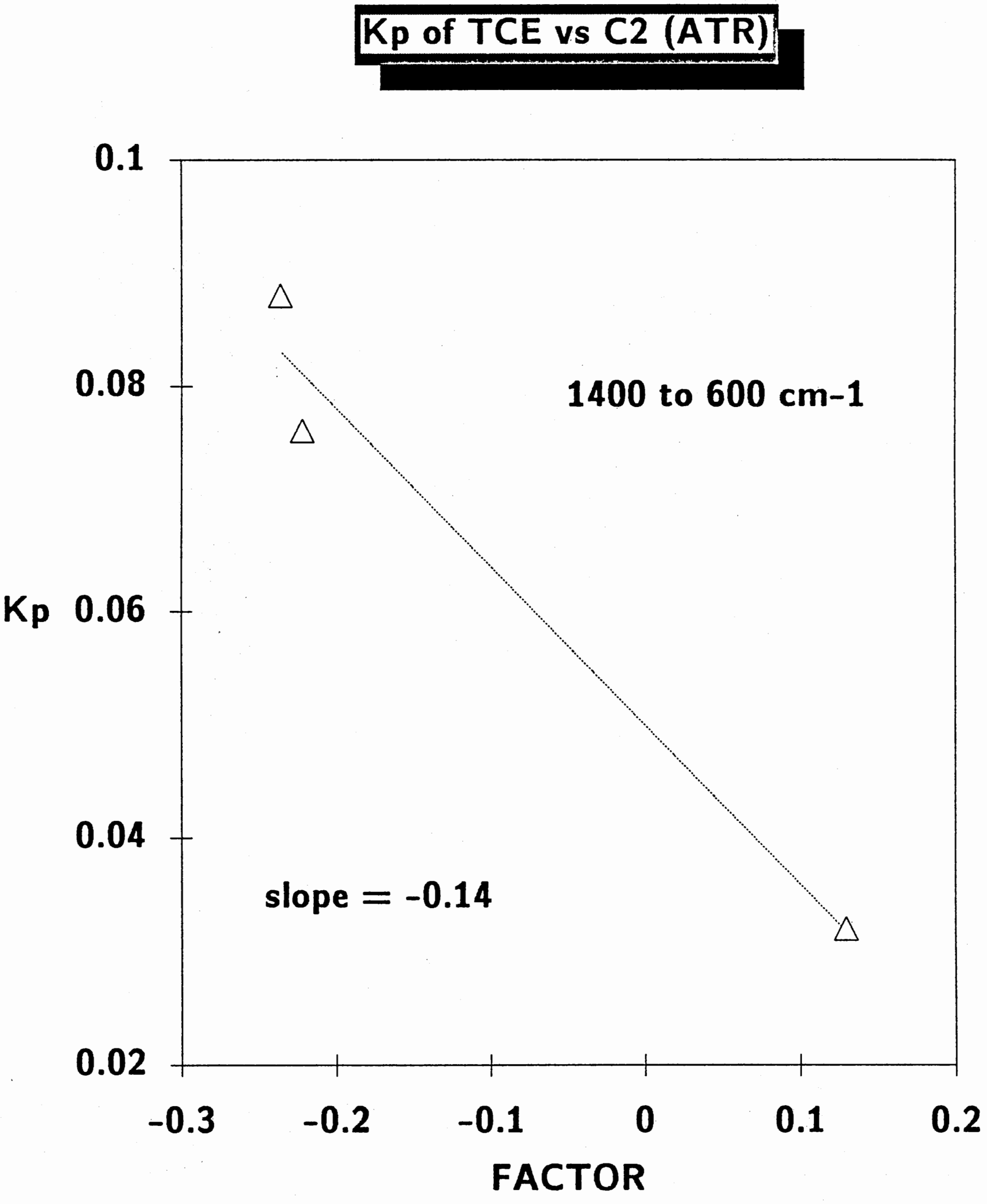


Figure 5b. Kp of TCE vs loadings on abstract factor C2 for EPA6, EPA8, and EPA9

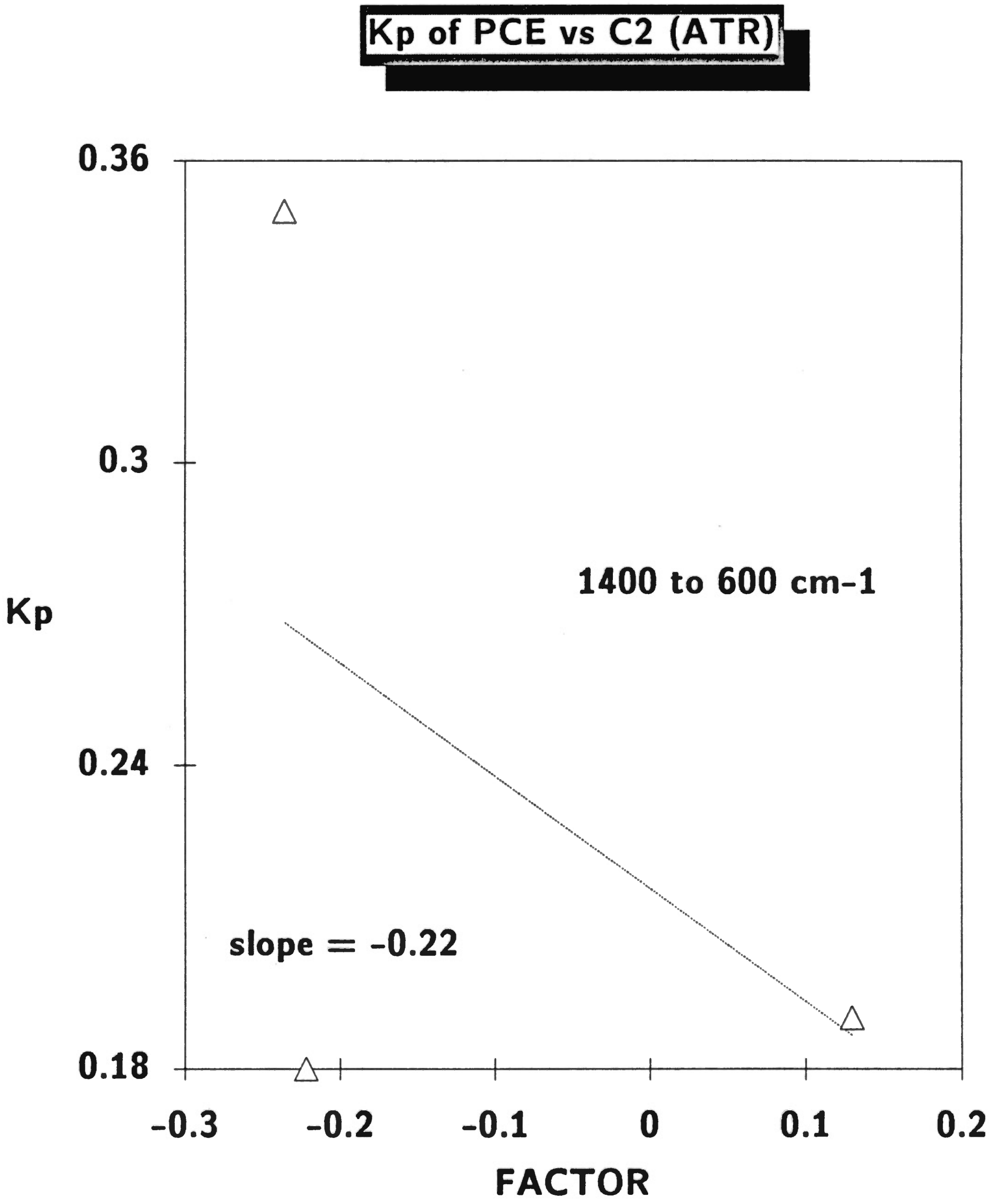


Figure 5c. Kp of PCE vs loadings on abstract factor C2 for EPA6, EPA8, and EPA9

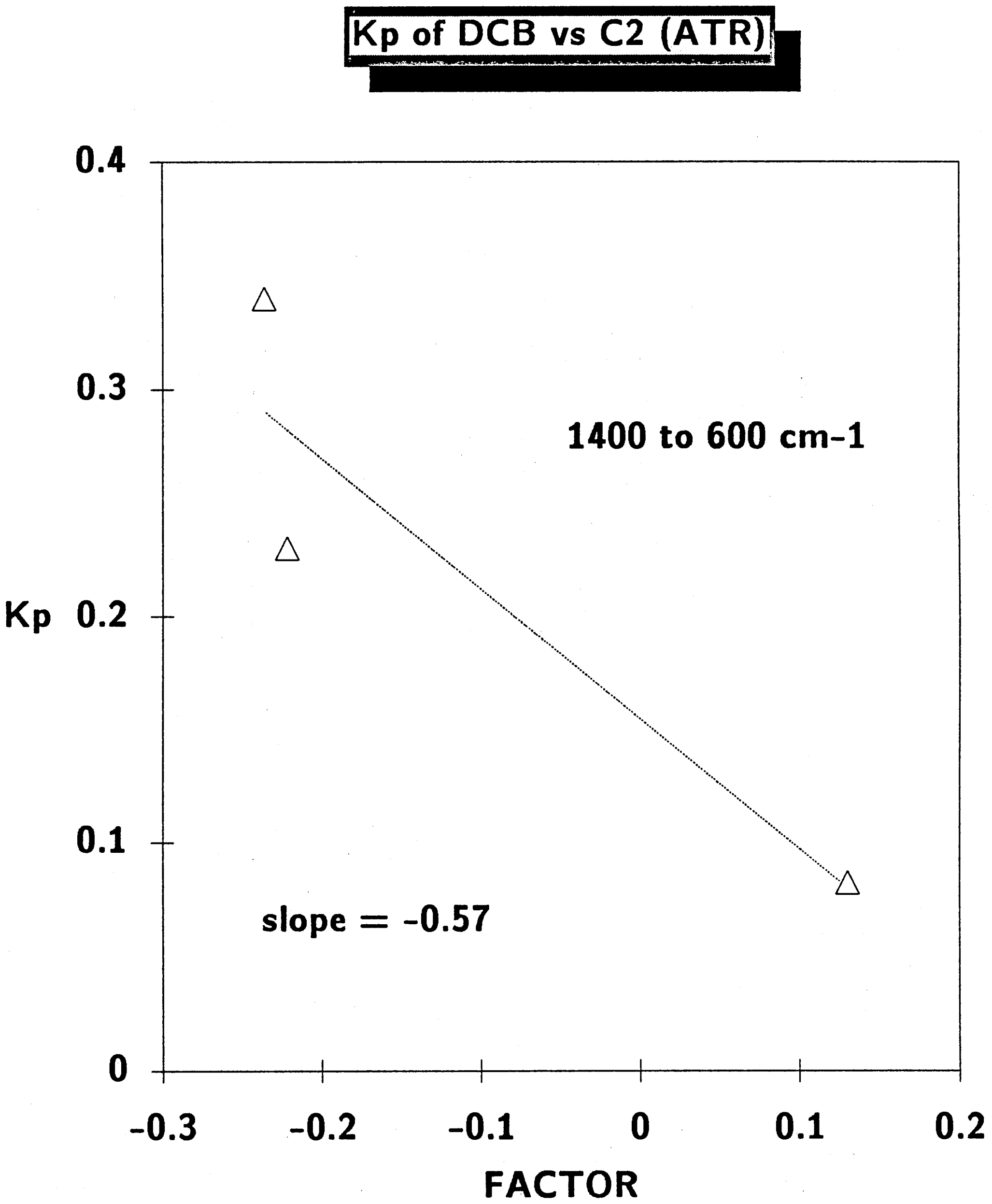


Figure 5d. Kp of DCB vs loadings on abstract factor C2 for EPA6, EPA8, and EPA9

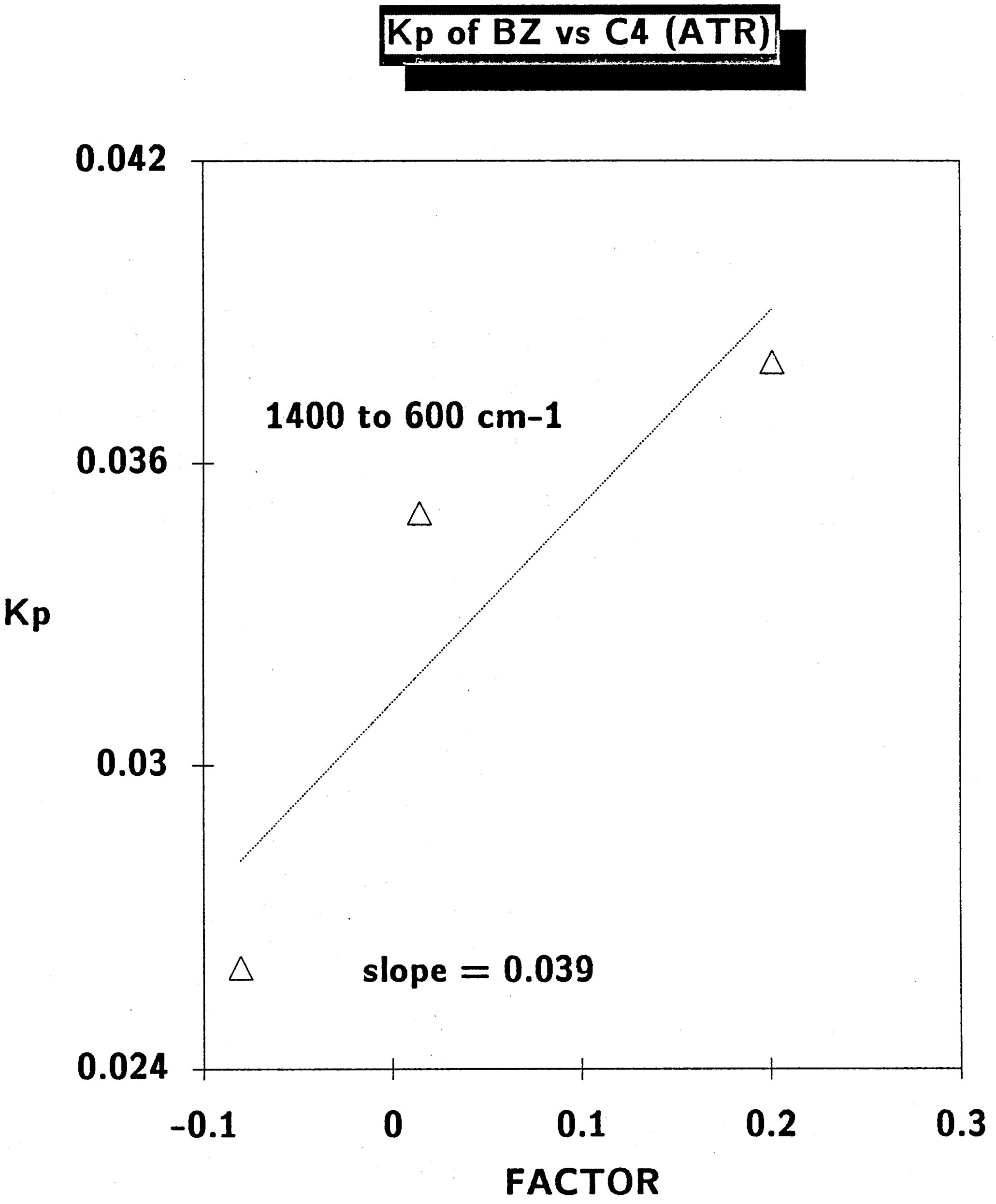


Figure 6a. Kp of BZ vs loadings on abstract fator C4 for EPA6, EPA8, and EPA9

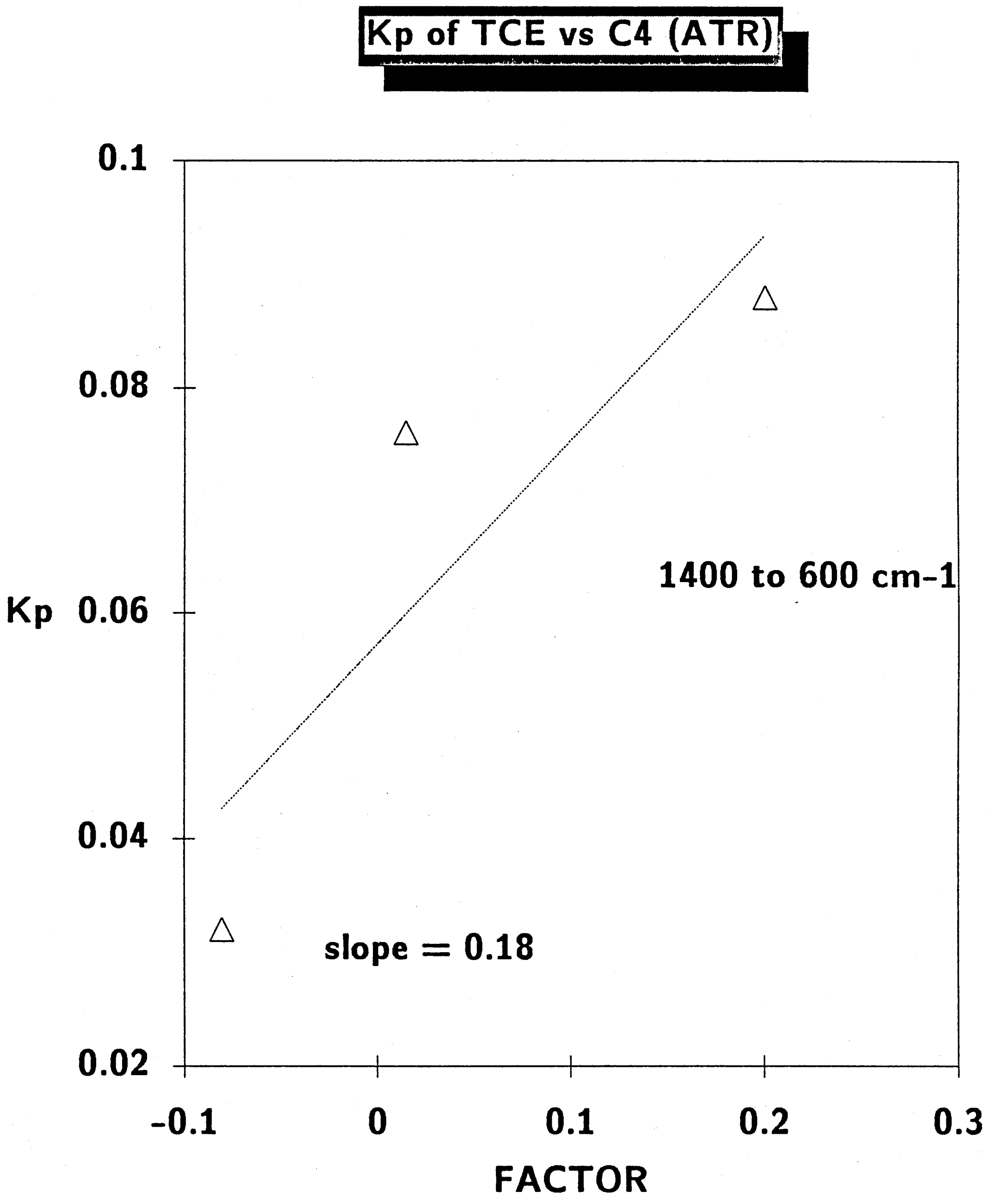


Figure 6b. Kp of TCE vs loadings on abstract factor C4 for EPA6, EPA8, and EPA9

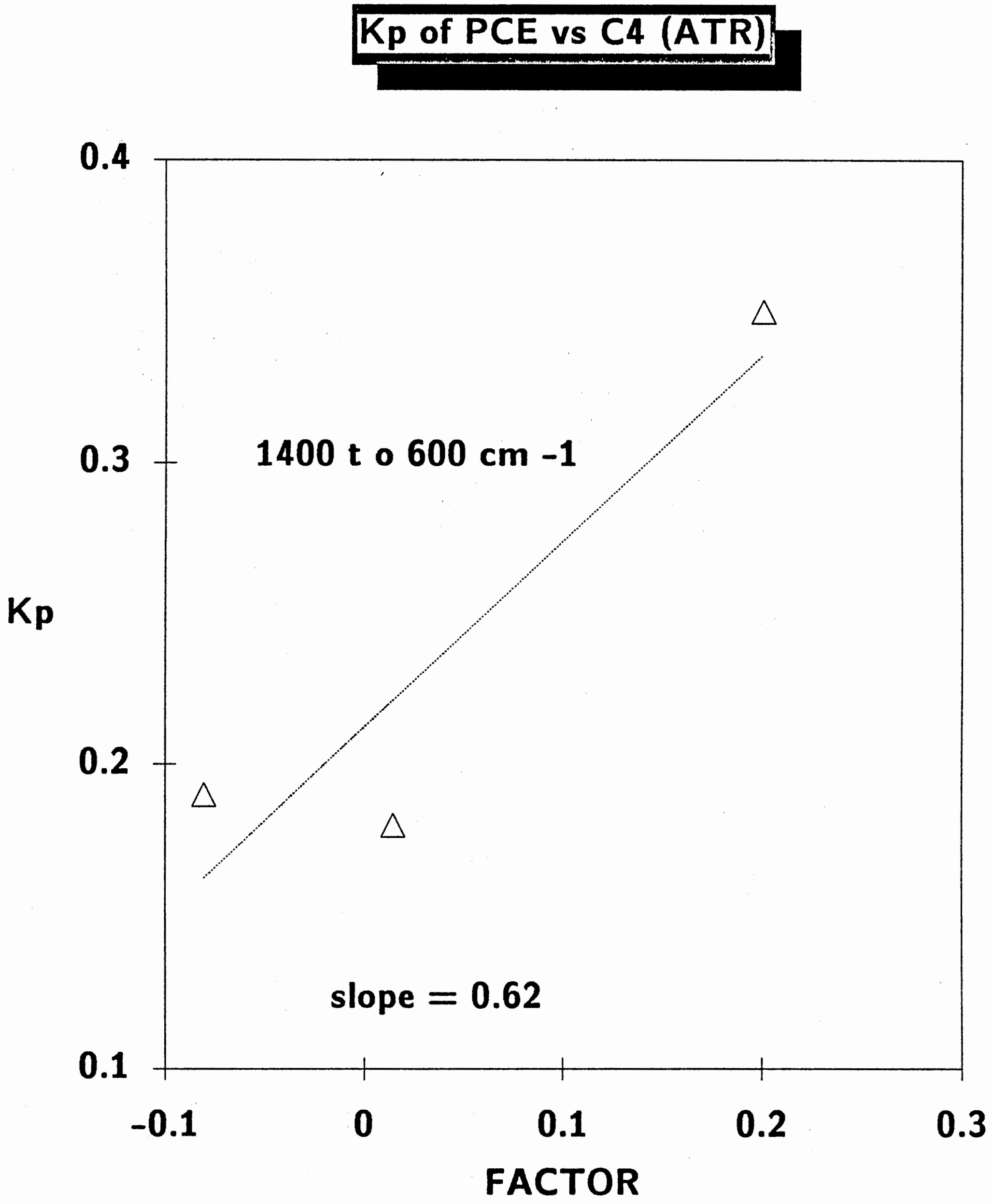


Figure 6c. Kp of PCE vs loadings on abstract factor C4 for EPA6, EPA8, and EPA9

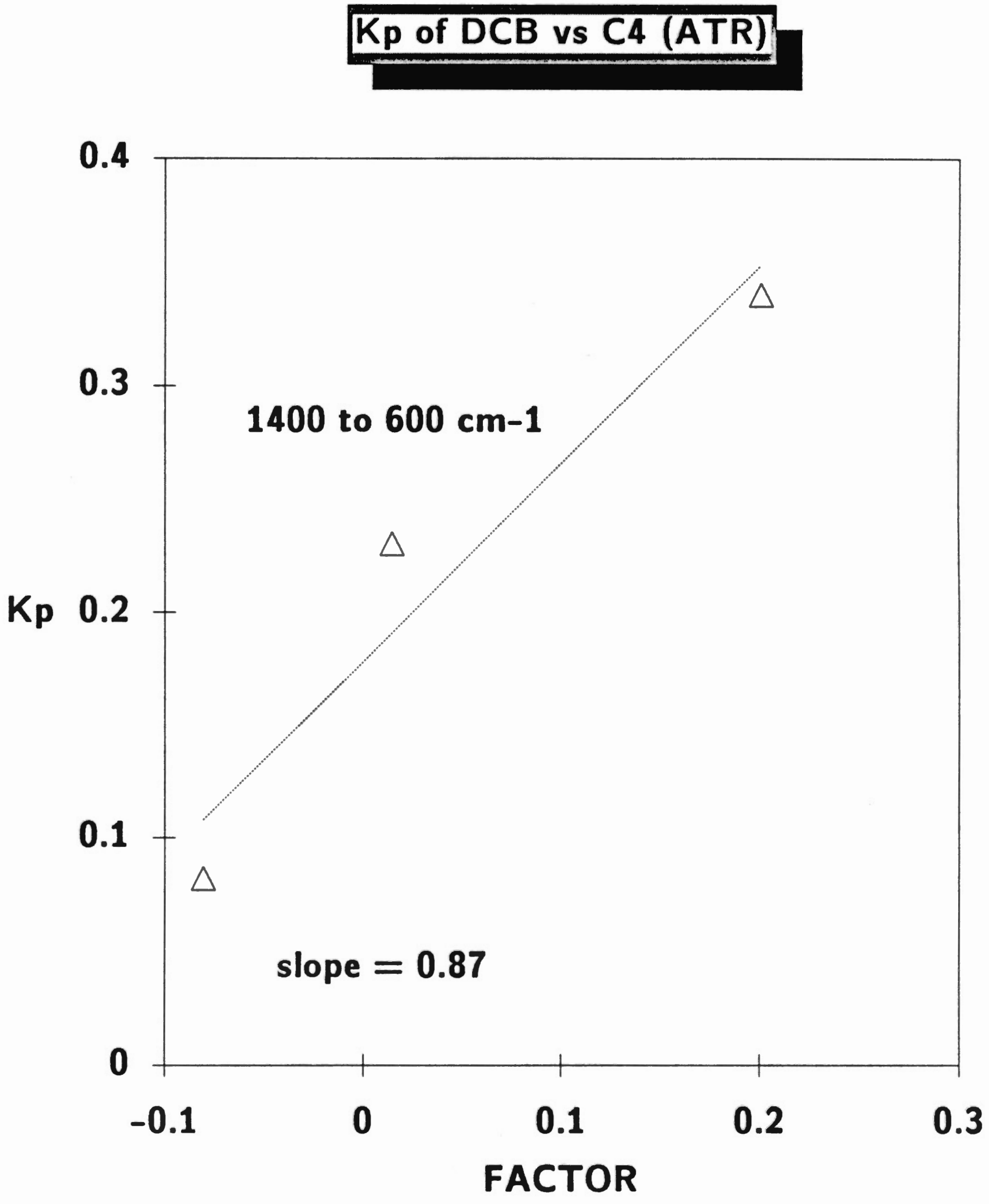


Figure 6d. Kp of DCB vs loadings on abstract factor C4 for EPA6, EPA8, and EPA9

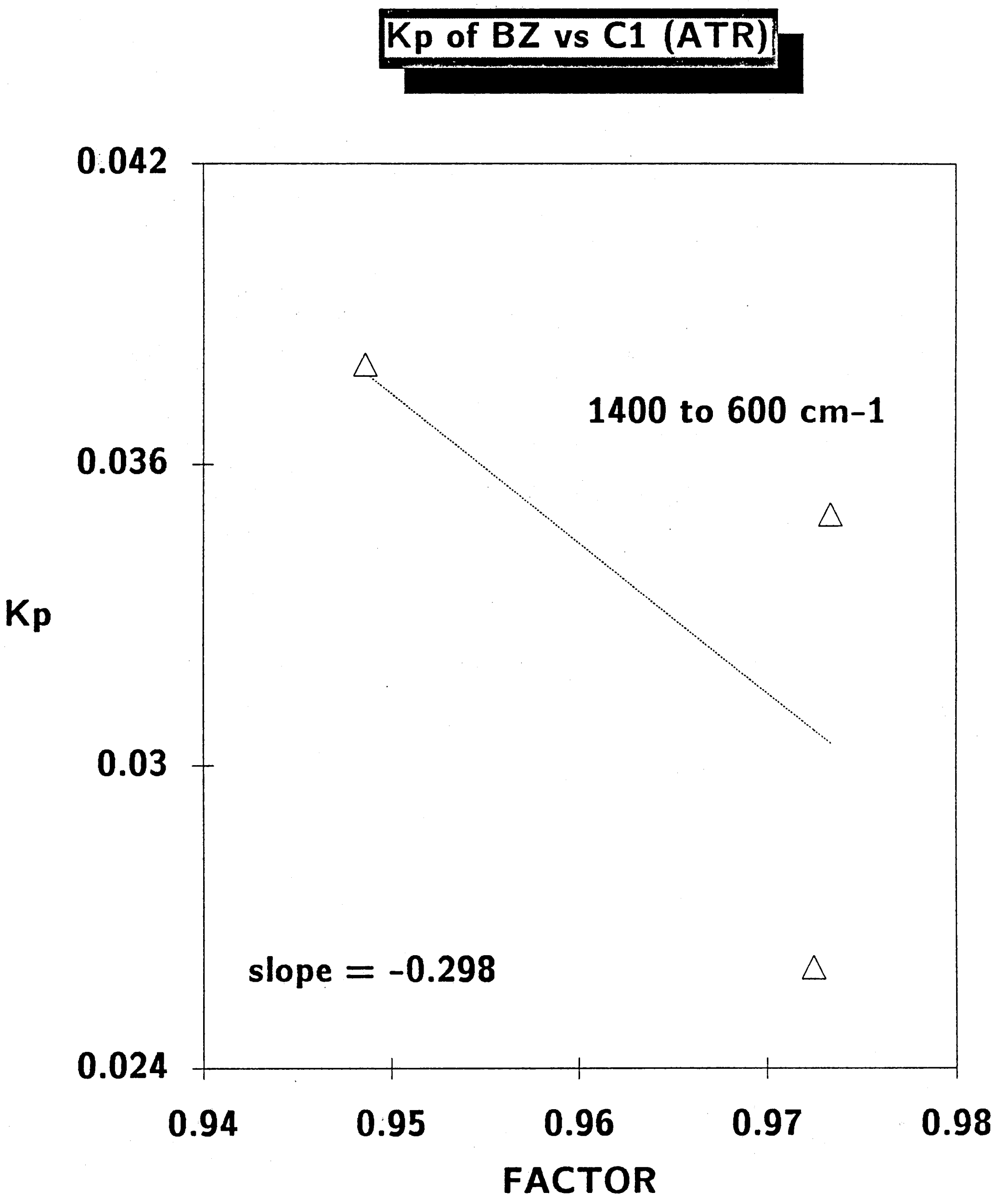


Figure 7a. Kp of BZ vs loadings on abstract factor C1 for EPA6, EPA8, and EPA9

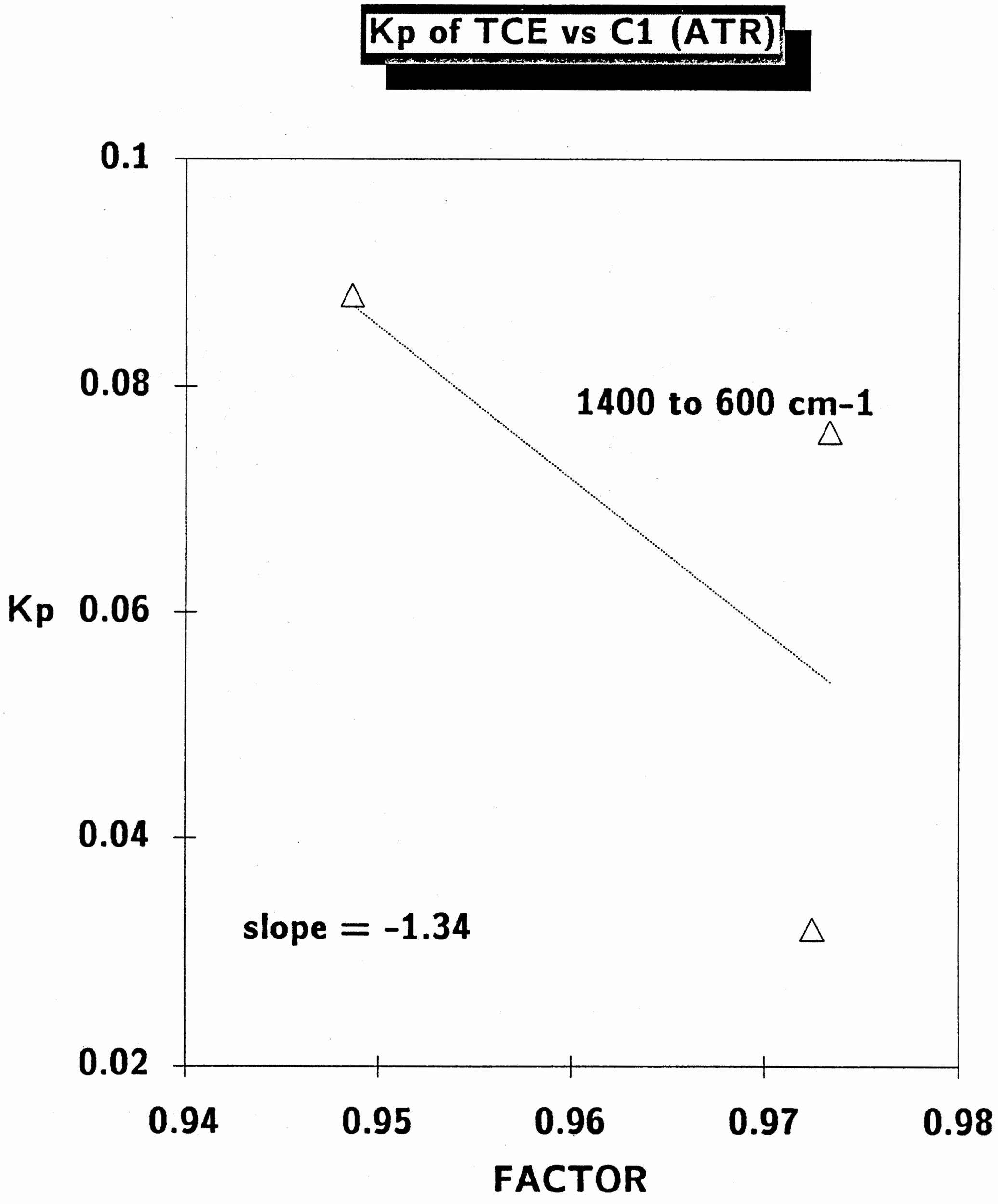


Figure 7b. Kp of TCE vs loadings on abstract factor C1 for EPA6, EPA8, and EPA9

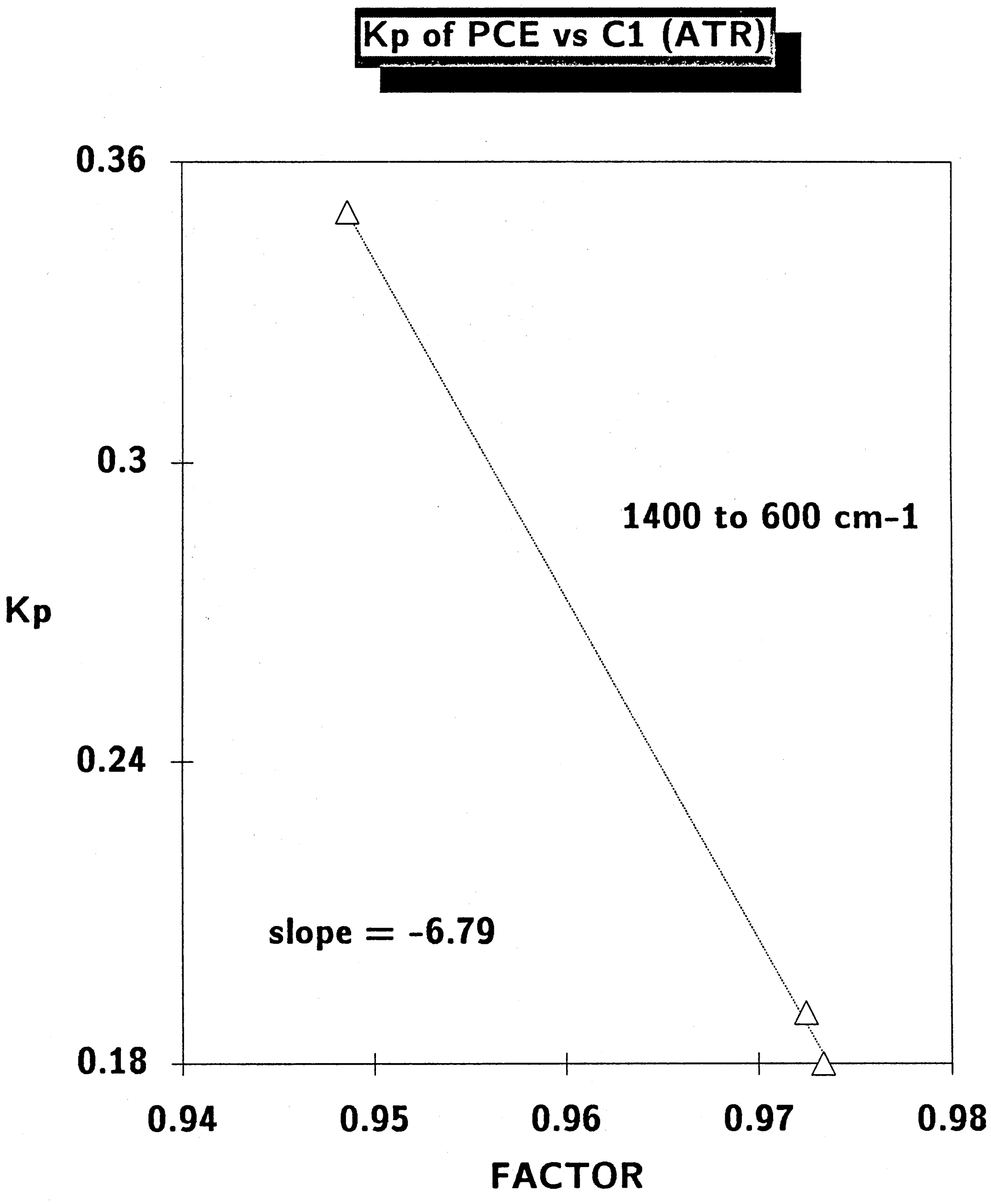


Figure 7c. Kp of PCE vs loadings on abstract factor C1 for EPA6, EPA8, and EPA9

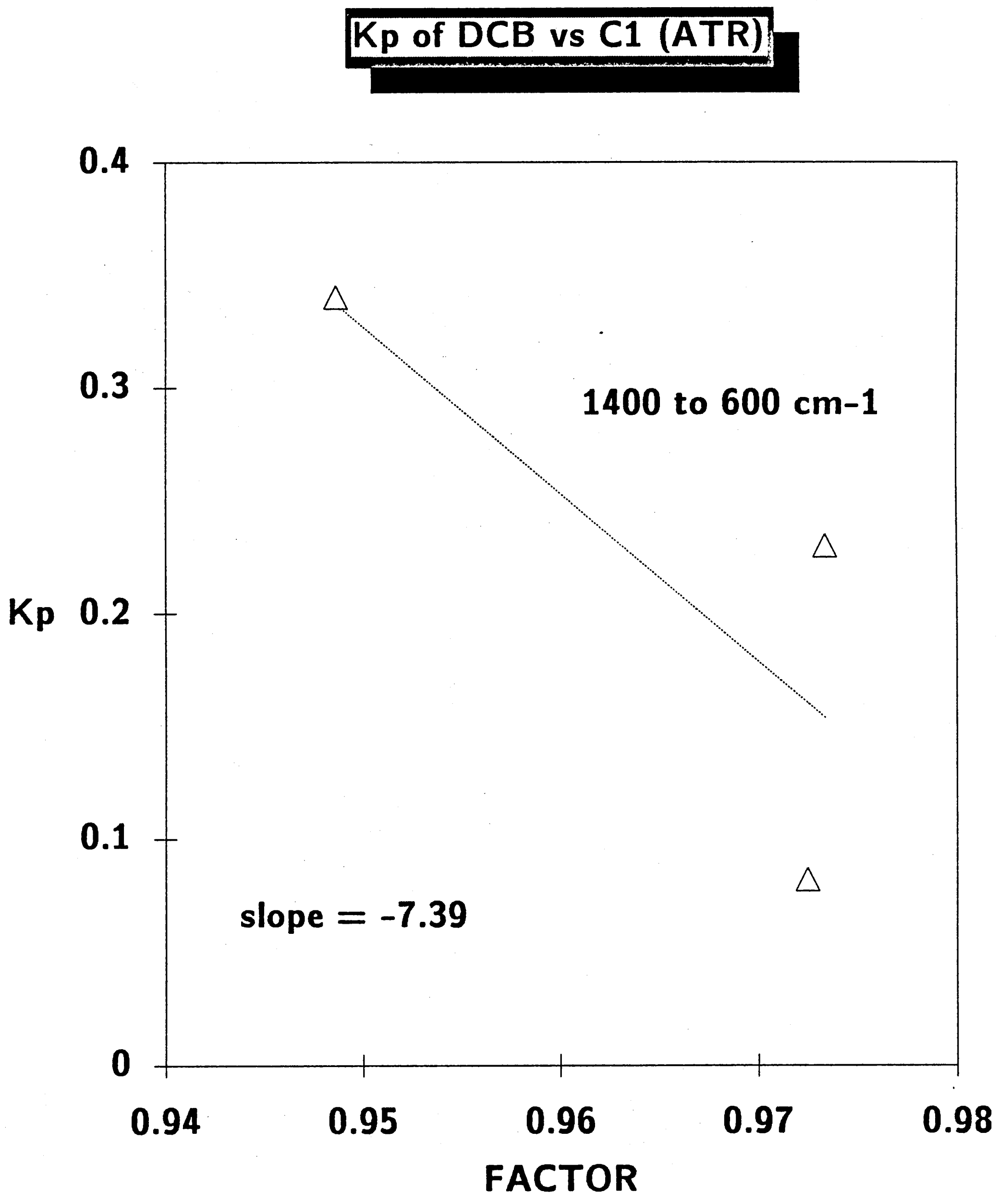


Figure 7d. Kp of DCB vs loadings on abstract factor C1 for EPA6, EPA8, and EPA9

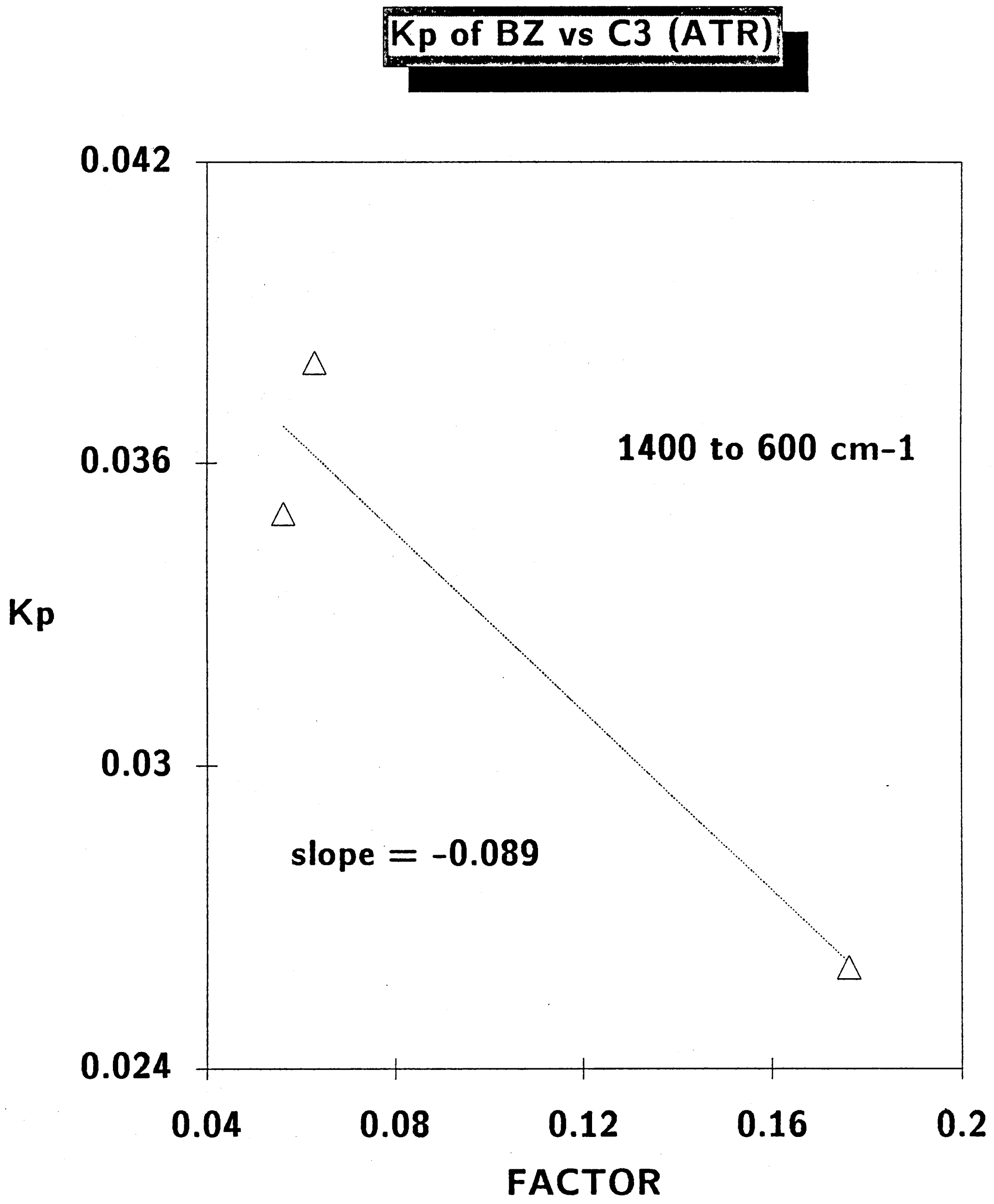


Figure 8a. Kp of BZ vs loadings on abstract factor C3 for EPA6, EPA8, and EPA9

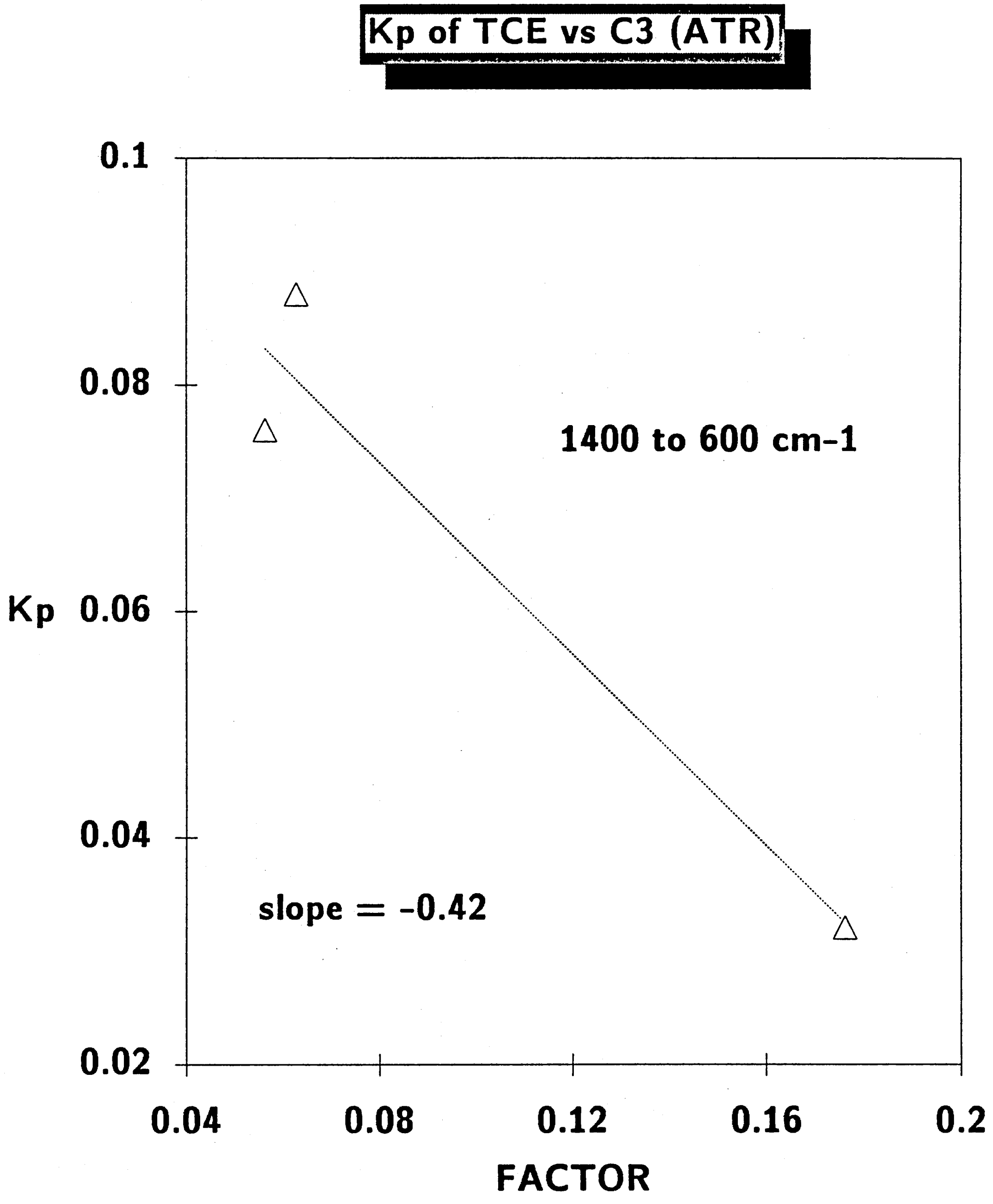


Figure 8b. Kp of TCE vs loadings on abstract factor C3 for EPA6, EPA8, and EPA9

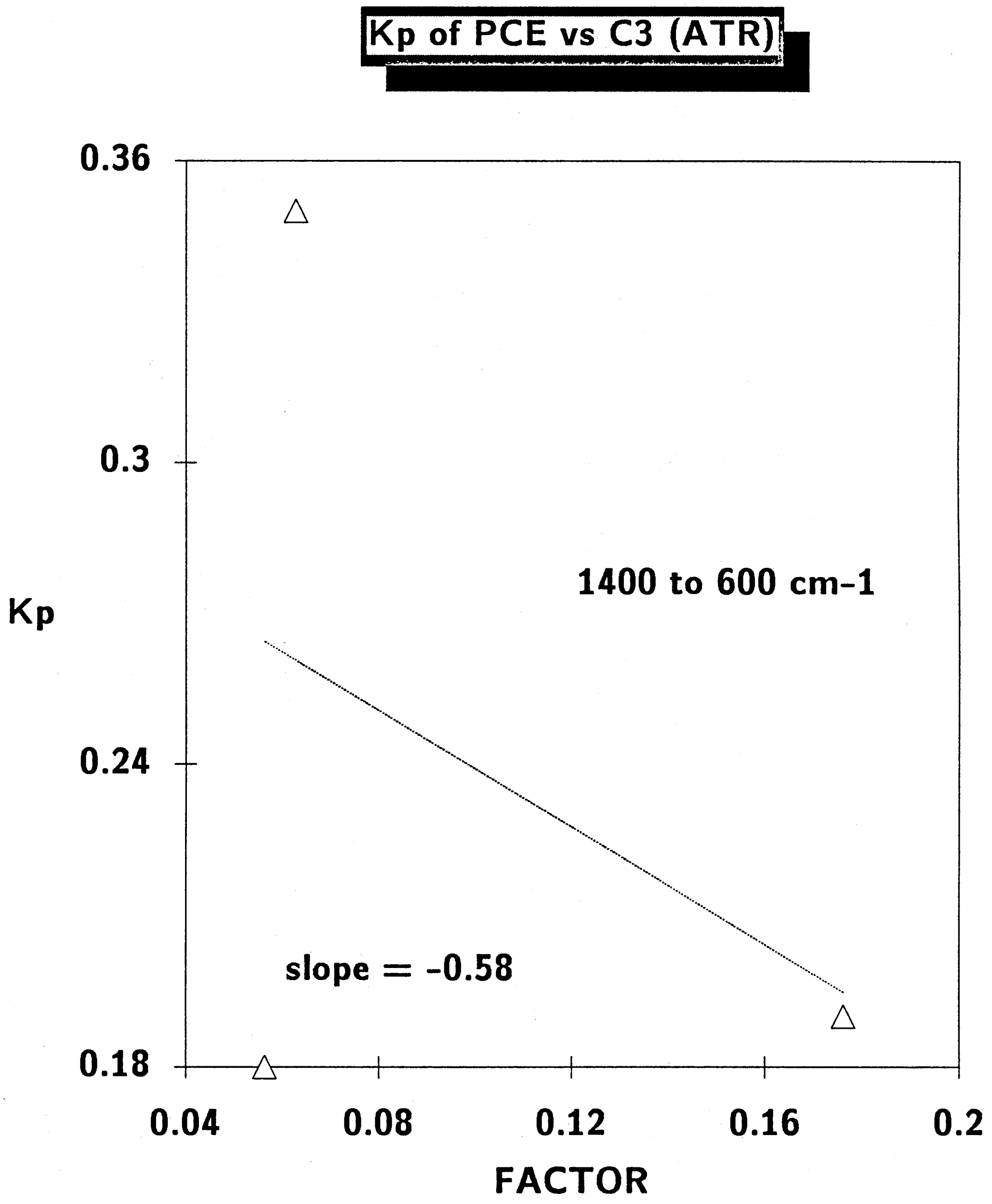


Figure 8c. Kp of PCE vs loadings on abstract factor C3 for EPA6, EPA8, and EPA9

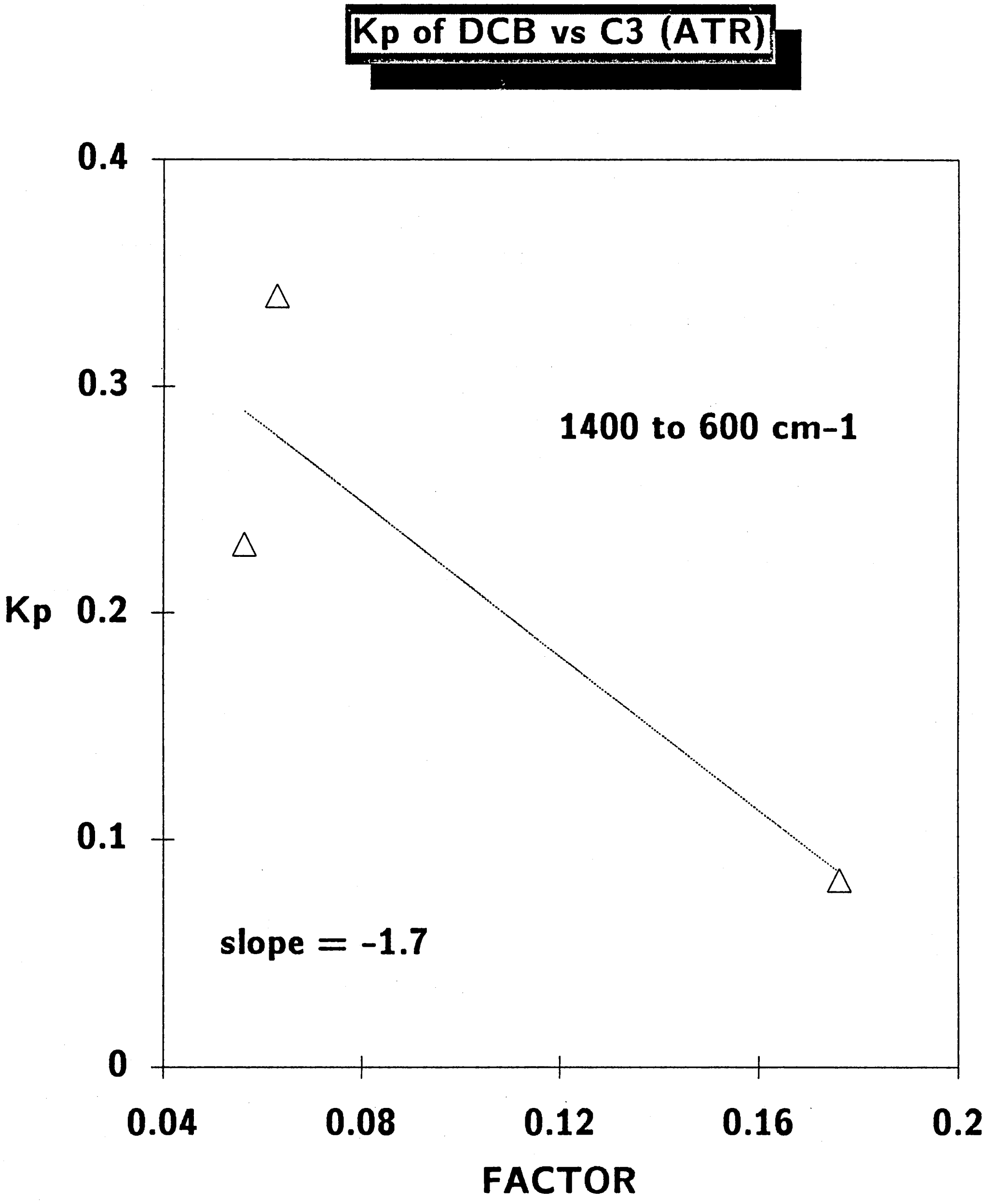


Figure 8d. Kp of DCB vs loadings on abstract factor C3 for EPA6, EPA8, and EPA9

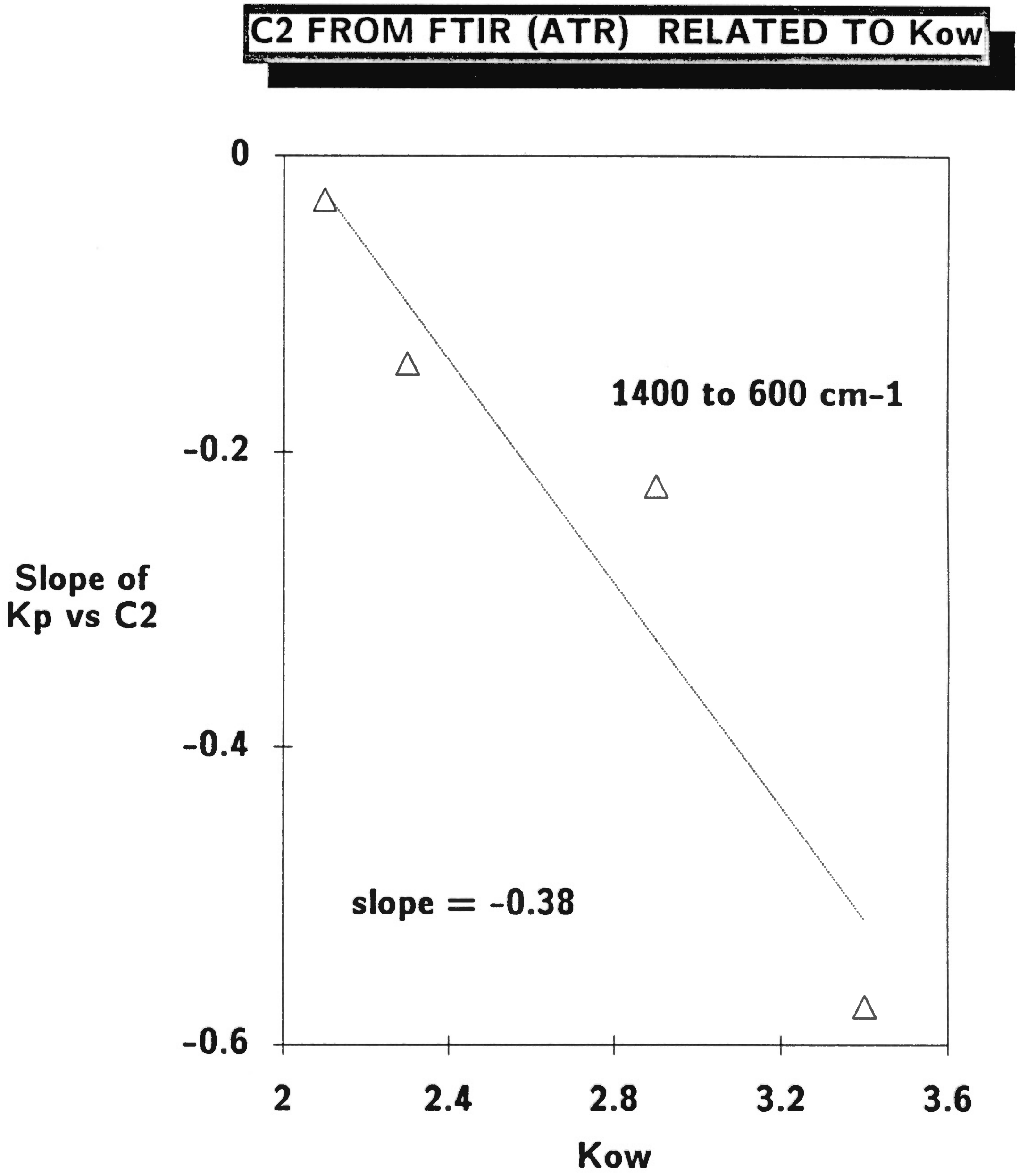


Figure 9a. Slope of Kp vs loading on abstract factor C2 VS Kow for the four solvents. (BZ, TCE, PCE, and DCB)

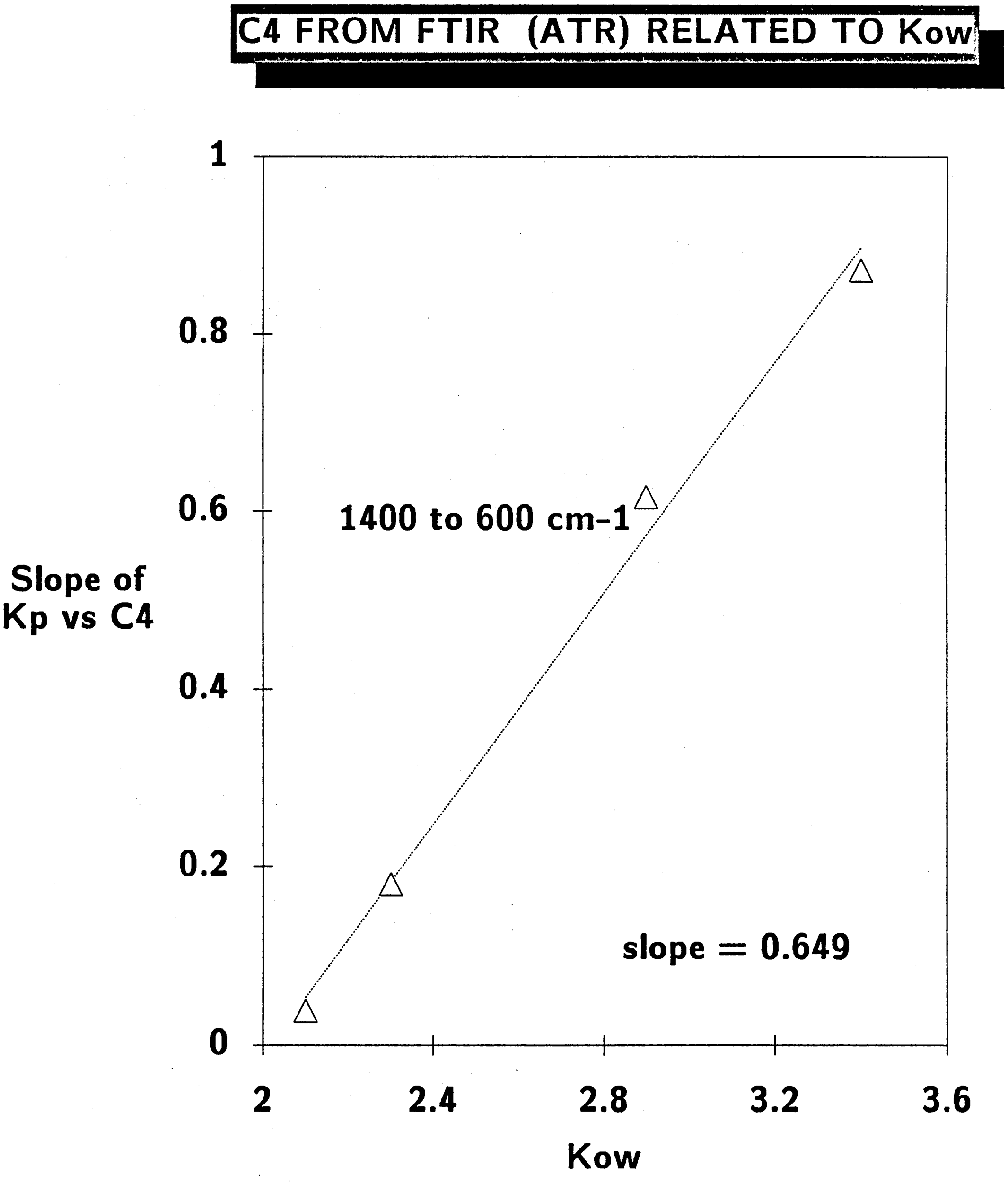


Figure 9b. Slope of Kp vs loading on abstract factor C4 VS Kow for the four solvents. (BZ, TCE, PCE, and DCB)

Correlation of Factors

There exists a rationale for the fact that only some of the factors seem to exhibit high correlation for the loadings of the soils on the different factors. It must be that the abstract factors 2 and 4 indicate the presence of physical factors which are responsible for pollutant retention in these samples. Although the correlation from the abstract factors 1 and 3 is not as good as that from abstract factors 2 and 4, it may well be that factors C1 and C3 indicate the presence of some low concentration causative agent responsible for pollutant retention in soils. Factors 1 through 4 are abstract factors, 1 indicating the most predominant component and 4 the least predominant component in the "average" of the set of ten soils. Since the aquifer minerals, from XRD studies appear to be predominantly quartz, abstract factors C2 through C4 may well correlate with the presence of various clay and carbonate components, in small concentrations, on the surface of the parent quartz component. Since useful correlations between abstract factors C2 and C4, and to a lesser extent C1 and C3 are observed, these components must play a very important role in pollutant sorption by aquifer minerals under conditions of low total organic carbon content (TOC).

On the basis of these results, it is seen that it may be possible to obtain the offset and slope of a straight line which reliably describes the relationship between Kp and the loading on factors 2 and 4 for any soil, for a given pollutant and its Kow. Further, it is possible to obtain loadings on those factors for a given low TOC soil using FTIR-ATR spectral analysis and factor compression. From this the Kp for the pollutant of interest on the soil under examination can be inferred. Thus by using a standard experimental technique - FTIR-ATR spectroscopy - and factor analysis, it is possible to obtain the Kp value of any pollutant on any low TOC soil. The time required from start to finish for this process is about 30 minutes. This may be compared with current methodology which involves time consuming column chromatography on large quantities of aquifer minerals.

In spite of the excellent results observed so far, the generalization of these results is unwise because of the limited number of samples studied. The dearth of measured Kp values imposes a severe constraint on the extensive application of these methods to study pollutant - soil interactions. The Environmental Protection Agency must enhance its scope of study to obtain more extensive sets of data, and make full use of this promising technique.

Conclusion

The conclusions based on these results are twofold. Firstly, the excellent spectral data obtained using FTIR-ATR as opposed to conventional pellet transmission spectroscopy (see figures 2a and 2b, and appendix C), justifies the use of this method for the infrared studies of aquifer minerals. We see that although the KBr pellet transmission spectrum of the sample in fig. 2a is good, the ATR spectrum of the same sample (fig. 2b) is even better under these circumstances. This is further supported by the fact that the use of FTIR-ATR minimizes the effects of particle size on the spectra. Using ATR, it is possible to record spectra of samples containing particles as big as 50 - 60 microns. This is not possible with pellet transmission spectroscopy. ATR is cost effective in that, by the use of a simple reflection element, older dispersive instruments may also be used to obtain infrared spectra. Moreover, ATR can be invaluable in cases where extremely small quantities of sample are available. Unlike pellet transmission spectroscopy, in which the sample is lost, ATR allows the reuse of the same sample for other studies. The problems associated with the Christiansen effect can also be of a serious nature in pellet transmission and FTIR-PAS spectroscopy (56). All the above stated advantages make ATR the method of choice for the analysis of aquifer minerals.

Secondly, the combination of FTIR-ATR spectroscopy with factor analysis provides us with a reasonably reliable and useful method to analyse pollutant - solvent systems for possible relationships that may eventually lead to the development of a useful relationship between pollutant Kow and low TOC soil Kp. That such a relationship between factor analyzed FTIR-ATR spectra and aquifer mineral sorption potential does exist is now certain, based on the results obtained with the limited set of data studied so far in this work. The four abstract factors generated by the factor analysis adequately describe the set of ten aquifer minerals studied. It is more than likely that these four abstract factors are indicative of physical components present in the aquifer minerals. This must be explored further. One direction to proceed would be to use procedures similar to automated spectral isolation of Lin and Liu (54) and Lin and Lin (55) as outlined earlier for the determination of the basic physical factors from the four defining abstract factors. This must be suitably supplemented by the EPA through provision of more Kp and Kow values for a wider variety of soil - pollutant systems. Successful spectral isolation using methods such as ASI has never been achieved before with a system as degenerate as the clay system. If, through the application of infrared spectroscopy and factor analysis this can be achieved the rewards would be enormous. This must be the catalyst that spurs more research in this environmentally important research area.

BIBLIOGRAPHY

1. Grim, R.E. Clay Minerology, McGraw-Hill, New York, New York (1953).

2. Mackenzie, R.C. and Mitchell, B.D., Clay Minerology Earth Sci.Revs., 2:47-91 (1966).

3. Theng, B.K.G., The Chemistry of Clay-Organic Reactions, Halsted Press, New York, New York (1974).

4. Faust, Samuel J., and Hunter, Joseph V., (Ed), Organic Compounds in Aquatic Environments, Marcel Dekker Inc., New York, New York (1971).

5. Legge, Allan H. and Krupa, Sagar V., (Ed), Air Pollutants and their Effects on The Terrestrial Ecosystem. Wiley Series in Advances in Environmental Science and Technology. 18:415-448, J.Wiley & Sons, New York, New York (1986).

6. Environmental Impact of Future Coal Production and Use in the EEC, A Report by Environmental Resources Limited, Graham & Trotman Limited, London, U.K. (1983).

7. Trace-Element Geochemistry of Coal Resource Development Related to Environmental Quality and Hazard, A Report by the Panel on The Trace Element Geochemistry of Coal Resource Development Related to Health, 37-39, 53-56, 60-61, 99-102, 103-116, National Academy Press, Washington, DC. (1980).

8. Magee, E.M., Bertrand, R.R., and Jahnig, C.E., Environmental Impacts and R&D needs in Coal Conversion, In Symposium Proceedings: Environmental Aspects of Fuel Conversion Technology, II. Report No. EPA 600/2-76-149. Washington, DC: Office of Research and Development, U.S. Environmental Protection Agency (1976). (loc. cit)

9. Hassett, John J., and Means, Jay C., Sorption Properties of Sediments and Energy Related Pollutants, Report No. EPA-600/3-80-041, U.S. Environmental Protection Agency, Environmental Research Laboratory, Office of Research and Development, Athens, Georgia, (1980).

10. Mackay, D.M., Freyberg, D.L., and Roberts P.V., A Natural Gradient Experiment On Solute Transport in a Sand Aquifer 1. Approach and Overview of Plume Movement, Water Reseources Research, 22:(13) 2017-2029 (1986).

11. Roberts, Paul V., Goltz, Mark N., and Mackay, Douglas M., A Natural Gradient Experiment on Solute Transport in a Sand Aquifer 3. Retardation Estimates and Mass Balances for Organic Solutes, Water Resources Research, 22:(13) 2047-2058 (1986).

12. Inglis, Mark J., Matisoff, Gerald., and Kelly, Walton Ross., Pollutant Transport in a Shallow Unconfined Aquifer in Perry, Ohio, Environ. Geol. Water Sci., 8:(4) 237-245 (1986).

13. Roy, W.R., and Griffin, R.A., Mobility of Organic Solvents in Water-Saturated Soil Materials, Environ. Geol. Water Sci., 7:(4) 241-247 (1985).

14. Piwoni, Marvin D., Behavior of Contaminants in the Subsurface - Factors Influencing Pollutant Fate in Soils. RSKERL, Ada, Oklahoma.

15. Curtis, Gary P., Roberts, Paul V., and Reinhard, Martin., A Natural Gradient Experiment on Solute Transport in a Sand Aquifer 4. Sorption of Organic Solutes and its Influence on Mobility, Water Resources Research, 22:(13) 2059-2067 (1986).

16. Banerjee, Pinaki., Piwoni, Marvin D., and Ebeid, Khawla., Sorption of Organic Contaminants to a Low Carbon Subsurface Core, Chemosphere, 14:(8) 1057-1067 (1985).

17. Schwarzenbach, Rene P., and Westall, John., Transport of Nonpolar Organic Compounds from Surface Water to Groundwater. Laboratory

Sorption Studies, Environmental Science & Technology, 15:(11) 1360-1367 (1981).

18. Piwoni, M.D., and Banerjee, P., Sorption of Volatile Organic Solvents from Aqueous Solution onto Subsurface Solids, Submitted to J.Contam.Hydrol, 4/86.

19. Rockley, M.G., Rockley, N.L., and Piwoni, M.D., Report to the U.S. Environmental Protection Agency, Unpublished (1987).

20. Rockley, M.G., Rockley, N.L., and Piwoni, M.D., FTIR Characterization of Aquifer Minerals, Report to the U.S. Environmental Protection Agency, Unpublished (1988).

21. Farmer, V.C., and Russell, J.D., The infra-red spectra of layer siilcates, Spectrochimica Acta, 20: 1149-1173 (1964).

22. Swoboda, A.R., and Kunze, G.W., Infrared Techniques for studying the Adsorption of Volatile Vapors by Clays, Soil Science, 101:(5) 373-377 (1966).

23. Farmer, V.C., and Russell, J.D., Infrared Adsorption Spectrometry in Clay Studies, Clays and Clay Minerals, 15: 121-142 (1967).

24. Olejnik, S., Aylmore, L.A.G., Posner, A.M., and Quirk, J.P., Infrared Spectra of Kaolin Mineral-Dimethyl Sulfoxide Complexes, The Journal of Physical Chemistry, 72:(1) 241-249 (1968).

25. Estep, Patricia A., Kovach, John J., Hiser, Arthur L., and Karr, Clarence Jr., Characterization of Carbonate Minerals in Oil Shales and Coals by Infrared Spectroscopy, Friedel (Ed), Spectrometry of Fuels, Chapter 18: 228-247 (1970).

26. Rouxhet, P.G., Samudacheata, Ngo., Jacobs, H., and Anton, O., Attribution of the OH Stretching Bands of Kaolinite, Clay Minerals, 12:171-179 (1977).

27. Sposito, Garrison., Prost, R., and Gautlier, J.-P., Infrared Spectroscopic Study of Adsorbed Water

on Reduced-Charge Na/Li- Montmorillonites, Clays and Clay Minerals, 31:(1) 9-16 (1983).

28. Lorprayoon, V., and Condrate, R.A. Sr., Infrared Spectra of Thiolane and Tetramethylene Sulfoxide Adsorbed on Montmorillonite, Clays and Clay Minerals, 31:(1) 43-48 (1983).

29. Rockley, M.G., FTIR Characterization of Aquifer Minerals, Unpublished (1985).

30. Griffiths, Peter R., and de Haseth, James A., Fourier Transform Infrared Spectrometry, Elving, P.J., and Winefordner, J.D., (Ed.), CHEMICAL ANALYSIS, A Series of Monographs on Analytical Chemistry and its Applications, A Wiley-Interscience Publication, 83: Preface. John Wiley & Sons, New York, New York (1986).

31. Rockley, M.G., FTIR Characterization of Aquifer Minerals, Unpublished (1985).

32. Rockley, M.G., Chem.Phys.Lett., 68: 455 (1979).

33. Rockley, M.G., Appl.Spectrosc., 34: 405 (1980).

34. Rockley, M.G., Richardson, H.H., and Davis, D.M., Fourier Transformed Infrared Photoacoustic Spectroscopy, the Techniques and its Applications, Ultrasonics Symposium Proceedings, 649-651 (1980).

35. Rockley, N.L., and Rockley, M.G., The FT-IR Analysis by PAS and KBr Pellet of Cation-Exchanged Clay Mineral and Phosphonate Complexes, Appl. Spectrosc., 41:(3) 471-475 (1987).

36. Rockley, M.G., Rockley, N.L., and Piwoni, M.D., FTIR Characterization of Aquifer Minerals, Unpublished (1987).

37. Miller, J.G., and Oulton, T.D., Some Effects of Grinding Kaolinite with Potassium Bromide, Clays and Clay Minerals, 20: 389-390 (1972).

38. Harrick, N.J., Internal Reflection Spectroscopy, John Wiley and Sons, New York, New York (1967).

39. Mielczarski, Jerzy A., Denca, Andrzej., and

Strojek, Jerzy W., *Application of Attenuated Total Reflection Spectroscopy to the Characterization of Coal,* Appl. Spectrosc., 40:(7) 998-1004 (1986).

40. Malinowski, Edmund R., and Howery, Darryl G., *Factor Analysis in Chemistry,* John Wiley & Sons, New York, New York (1980).

41. Fredricks, Peter M., Lee, James B., Osborn, Paul R. and Swinkels, Dom A.J., *Materials Characterization Using Factor Analysis of FT-IR Spectra. Part 1: Results,* Appl. Spectrosc., 39:(2) 303-310 (1985).

42. Fredricks, Peter M., Lee, James B., Osborn, Paul R. and Swinkels, Dom A.J., *Materials Characterization Using Factor Analysis of FT-IR Spectra. Part 2: Mathematical and Statistical Considerations.,* Appl. Spectrosc., 39:(2) 311-316 (1985).

43. Pfluger, C.E., *X-Ray Diffraction.,* Analytical Chemistry, 42:(5) 317R-324R (1970).

44. Davis, Briant L., and Walawender, Michael J., *Quantitative minerological analysis of granitoid rocks: a comparision of X-ray and optical techniques.,* American Minerologist, 67: 1135-1143 (1982).

45. *Perkin-Elmer Model 1710 Infrared Fourier Transform Spectrometer, Operator's Manual,* Copyright by Perkin-Elmer Ltd., Beaconsfield, Buckinghamshire, England (1984).

46. *Harrick Prism Liquid Cell Data Sheet # 10,* Harrick Scientific Corporation, Ossining, New York (1983).

47. *Smart Cablemaker Model SC821PLUS, User Manual,* IQ Technologies, Inc., Bellevue, WA (1987).

48. Troy, Douglas A., *Complete C Language Programming for the IBM PC,* Little, Brown and Company, Boston, MA (1986).

49. Kernighan, Brian W., and Plauger, P.J., *The Elements of Programming Style,* Second Edition, McGraw-Hill Book Company, New York, New York (1978.

50. Bolon, Craig, Mastering C, Sybex Inc., Berkeley, California (1986).

51. Kernighan, Brian W., and Ritchie, Dennis M., The C Programming Language, Prentice-Hall of India Private Limited, New Delhi, India. (1986).

52. Biggerstaff, Ted J., Systems Software Tools, Prentice-Hall, Englewood Cliffs, NJ. (1986).

53. Microsoft C Compiler, Version 4.0, Microsoft Corporation, Bellevue, WA. (1985).

54. Lin, C.H., and Liu, S.C., J. Chinese Chem. Soc., 25: 167 (1978). (loc. cit)

55. Lin, C.H., and Lin, L.C., Proc. Natl. Sci. Counc., ROC 3: 1 (1978). (loc. cit)

56. Farmer, V.C, and Russell, J.D., Effects of particle size on the vibrational frequencies of layer silicates., Spectrochimica Acta, 22: 389-398 (1965).

APPENDIX A

COMPUTER PROGRAMS

ORIGINAL FACTOR ANALYSIS PROGRAM

```
/*      This program performs a factor analysis of              */
/*      all dominant factors in a random set of spectra         */

#include "math.h"
#include "stdio.h"
#include "stdlib.h"
#define SPEC_SIZE 2300
#define RANK 12         /* Typical number of eigenvalues expected */
#define RANGE 400       /* Typical number of wavelengths analyzed */
#define NOISE 0.08      /* 1000:1 T mode noise at 50% transmission*/
#define DELTA_F 1.9294  /* frequency spacing in wavenumbers       */
#define HIGH_F 4814     /* Starting frequency in cm-1 of Spectra  */


float   predicted[RANGE],testvector[RANGE],risk[RANGE];
float   irinfo[SPEC_SIZE],datamatrix[RANGE][RANK];
float   covariancematrix[RANK][RANK];
float   runningmatrix[RANK][RANK],subtrahend[RANK][RANK];
float   abst_row[RANGE][RANK],eigenvalues[RANK],guess[RANK],guess2[RANK];
float   crosselement,improvement,eigenvectors[RANK][RANK];
double  amplitude,oldamplitude,residual;
int     elements,samples,result,i,j,k,l,row,column,passthru,sampcnt;
int     start_wlength,end_wlength,where_to_begin,position,space_size;
char    query,query2,fname[1][14],dateinfo[30],inputname[RANK][20],fdataname[20];
FILE    *fdata;
int     best_guesses[RANK],lowpossible;
char    stopit;

main()
{
FILE    *f9;
float   temp;
int     sampcnt;

result=system("cls");
printf("          Please Enter the DATE (any format) ...");
scanf("%s",dateinfo);
printf("\n\n\n  *******      ABSTRACT FACTOR ANALYTICAL REDUCTION    ******\n");
printf("                     Analysis Date .... %s\n",dateinfo);
printf("          Enter file name for data output ......");
scanf("%s",fdataname);
fdata=fopen(fdataname,"w");
fprintf(fdata,"\n\n\n  *******      ABSTRACT FACTOR ANALYTICAL REDUCTION    ******\n");
fprintf(fdata,"                     Analysis Date .... %s\n",dateinfo);
printf("          Enter the number of samples ..........");
scanf("%d",&samples);
printf("          Will data be from IRD files? (y/n) ...");
query=getche();
passthru=-1;
```

```
if ((query!='y') && (query!='Y'))
        {query2='n';
        printf("\n          Enter the number of elements/sample  ");
        scanf("%d",&elements);
        }
if ((query=='y') || (query=='Y'))
        {lowpossible=400+(RANGE*1.9294);
        printf("\n          Enter --integer-- for starting frequency in cm-1 ... ");
        do
          {printf("\n          Value must be greater than %4d cm-1. Value = ",lowpossible);
          scanf("%d",&where_to_begin);
          }
        while (where_to_begin<lowpossible);
        printf("   Starting frequency in cm-1 = %5d \n",where_to_begin);
        printf("   %4d points used at a spacing of %6.4f cm-1",RANGE,DELTA_F);
        printf("\n");
        fprintf(fdata,"   Starting frequency in cm-1 = %5d \n",where_to_begin);
        fprintf(fdata,"   %4d points used at a spacing of %6.4f cm-1",RANGE,DELTA_F);
        fprintf(fdata,"\n");
        start_wlength=(HIGH_F-where_to_begin)/DELTA_F;
        end_wlength=start_wlength+RANGE;
        elements=RANGE;
        }
f9=fopen("BOOKDATA","r");
for (sampcnt=0;sampcnt<samples;++sampcnt )
        {printf("\n");
        if ((query=='y') || (query=='Y'))
                {printf("Name of file for sample # %3d = ",sampcnt);
                scanf("%s",fname);
                printf("\n");
                get_irinfo(fname,irinfo,SPEC_SIZE);
                put_irinfo_into_data(datamatrix,sampcnt,irinfo,elements);
                }
        else
                {for (j=0;j<elements;++j )
                        {printf("Element # %3d = ",j);
                        fscanf(f9,"%f",&temp);
                        datamatrix[j][sampcnt]=temp;
                        printf("%6.3f ",temp);
                        }
                }
        }
fclose(f9);
find_the_covariance_matrix();
printf("\n\n                ****** EIGENVALUES and EIGENVECTORS ******");
fprintf(fdata,"\n\n                ****** EIGENVALUES and EIGENVECTORS ******\n");
do
     {passthru=passthru+1;
     guess_the_initial_vector();
     run_x_initial_guess();
     load_vector_into_eigenvectors();
     find_new_running_matrix();
     get_residual();
     }
while ((passthru<samples-1) );
display_errors();
calc_abstract_row_matrix();
if ((query!='y') && (query!='Y') && (query2!='y') && (query2!='Y'))
     disp_abst_row_matrix();
find_loadings();
fclose(fdata);
/*needle_search();*/
}       /* End of procedure MAIN                              */



find_the_covariance_matrix()
{
int     i,k,l;

for (i=0;i<samples;++i )
        {for (k=0;k<samples;++k )
                {covariancematrix[i][k]=0.0;
                for (l=0;l<elements;++l)
                  {crosselement=datamatrix[l][i]*datamatrix[l][k];
                  covariancematrix[i][k]+=crosselement;
                  }
                runningmatrix[i][k]=covariancematrix[i][k];
                }
        }
}      /*of the function to find the covariance matrix DtD      */


guess_the_initial_vector()
{
double  temp;
int     i;

for (i=0;i<samples;++i )
        {temp=1.0/samples;
        guess[i]=sqrt(temp);
        }
}      /*of the function to guess the initial vector            */


run_x_initial_guess()
{
float   posneg;

amplitude=0.0;
```

```
do
     {for (i=0;i<samples;++i )
        {crosselement=0.0;
        for (j=0;j<samples;++j )
               crosselement+=runningmatrix[i][j]*guess[j];
        guess2[i]=crosselement;
        }
     oldamplitude=amplitude;
     amplitude=0.0;
     for (i=0;i<samples;++i )
          amplitude+=guess2[i]*guess2[i];
     for (i=0;i<samples;++i )
          guess[i]=guess2[i]/(sqrt(amplitude));
     printf("\n");
     posneg=(1.00-(oldamplitude/amplitude));
     if (posneg<0)posneg=-posneg;
     improvement=posneg;
     locatxy(15,20);printf("     ");locatxy(1,20);
     printf("improvement = %6.4f\n",improvement);
     }
while (improvement>0.001);
eigenvalues[passthru]=sqrt(amplitude);
printf("\n");
printf("Amplitude -Eigenvalue lambda- = %7.3f\n",eigenvalues[passthru]);
fprintf(fdata,"Amplitude -Eigenvalue lambda- = %7.3f\n",eigenvalues[passthru]);
printf("That eigenvector is  ....");
fprintf(fdata,"That eigenvector is  ....");
for (i=0;i<samples;++i )
        {printf("%6.3f   ",guess[i]);
        fprintf(fdata,"%6.3f   ",guess[i]);
        }
printf("\n");
fprintf(fdata,"\n");
}             /*of the function to mult. running matrix by guess vector */


get_residual()
{
float temp;

residual=0.0;
for (i=0;i<samples;++i )
   {for (j=0;j<samples;++j )
        {temp=runningmatrix[i][j];
        residual+=temp*temp;
        }
   }
}              /*of the function to find the residual sum of sqrs of matrix*/
```

```
find_new_running_matrix()
{
for (i=0;i<samples;++i )
        {for (j=0;j<samples;++j )
                subtrahend[i][j]=0.0;
        }
for (i=0;i<samples;++i )
        {for (j=0;j<samples;++j )
                {subtrahend[i][j]+=(eigenvalues[passthru]
                *eigenvectors[passthru][i]*eigenvectors[passthru][j]);
                }
        }
for (i=0;i<samples;++i)
        {for (j=0;j<samples;++j)
                runningmatrix[i][j]-=subtrahend[i][j];
        }
}            /*of function to find new running matrix   */


load_vector_into_eigenvectors()
{
for (i=0;i<samples;++i )
        eigenvectors[passthru][i]=guess[i];
}             /*of function to load iterated guess into eigenvectors    */


display_errors()
{
double  temp;
float   re,ie,xe,ind,ind_test;
int     u,t,n,j,infinity;

if (samples>elements)  u=samples; else u=elements;
if (samples <elements)  t=samples; else t=elements;
n=passthru;
space_size=1;   /*increment by one until indicator minimizes*/
printf("\n\n               *******  ERROR ANALYSIS  *******\n");
printf("RE = real error\n");printf("XE = extracted error\n");
printf("IE = imbedded error\n");
printf("IND= indicator function\n");
printf("\nEigenvalue    RE        XE         IE         IND\n");
printf("------------------------------------------------\n");
fprintf(fdata,"\n\n               *******  ERROR ANALYSIS  *******\n");
fprintf(fdata,"RE = real error\n");printf("XE = extracted error\n");
fprintf(fdata,"IE = imbedded error\n");
fprintf(fdata,"IND= indicator function\n");
fprintf(fdata,"\nEigenvalue    RE        XE         IE         IND\n");
fprintf(fdata,"------------------------------------------------\n");
for (n=0;n<=passthru;++n)
        {infinity=0;
```

```
       re=0;ie=0;xe=0;
       for (j=n+1;j<t;++j)
            {re+=eigenvalues[j];
            xe+=eigenvalues[j];
            }
       if (t>n)
            {temp=(re/(u*(t-n)));
            re=sqrt(temp);
            ind=re/((t-n)*(t-n));
            if (n==0)
               ind_test=ind;
            else if (ind<=ind_test)
               ++space_size;
            }
       if (t<=n) infinity=1 ;
       temp=xe/(u*t);
       xe=sqrt(temp);
       temp=(re*re)-(xe*xe);
       ie=sqrt(temp);
       if (infinity==0)
            {printf("%10.5f %10.5f %10.5f %10.5f %10.5f\n",
               eigenvalues[n],re,xe,ie,ind);
            fprintf(fdata,"%10.5f %10.5f %10.5f %10.5f %10.5f\n",
               eigenvalues[n],re,xe,ie,ind);
            }
       else {printf("%10.5f   ***    %10.5f   ***         ***\n",eigenvalues[n],xe);
             fprintf(fdata,"%10.5f   ***    %10.5f   ***         ***\n",eigenvalues[n],xe);
            }
       }
}           /*of function to find the errors and display them   */



calc_abstract_row_matrix()
{
float   storage;
int     i,j,k,l;

for (i=0;i<=passthru;++i)
        {for (j=0;j<elements;++j )
                {storage=0.0;
                for (k=0;k<samples;++k )
                       storage+=(datamatrix[j][k]*eigenvectors[i][k]);
                abst_row[j][i]=storage;
                }
        }
}           /* function to calculate abstract row matrix        */


disp_abst_row_matrix()
```

```
{
int     i,j,k,l;

printf("\n    Abstract Row Matrix R  ");
printf("\n----------------------------\n");
fprintf(fdata,"\n    Abstract Row Matrix R  ");
fprintf(fdata,"\n----------------------------\n");
for (i=0;i<elements;++i)
        {for (j=0;j<=passthru;++j)
                {printf("%10.5f ",abst_row[i][j]);
                fprintf(fdata,"%10.5f ",abst_row[i][j]);
                }
        printf("\n");
        fprintf(fdata,"\n");
        }
printf("\n\n");
fprintf(fdata,"\n\n");
}         /*of function to display the abstract row matrix      */


find_loadings()
{
double  temp;
float   loading[RANK][RANK],amount;
int     i,j,k,l;

printf("\n\n                ",
       " ****** DATA COLUMN LOADINGS ONTO THE FACTORS ******");
printf("C=eigenvector (or factor) #\n");
printf("     C1       C2       C3       C4       C5");
printf("       C6       C7       C8       C9");
printf("       C10      C11      C12\n");
fprintf(fdata,"\n\n                ",
       " ****** DATA COLUMN LOADINGS ONTO THE FACTORS ******");
fprintf(fdata,"C=eigenvector (or factor) #\n");
fprintf(fdata,"     C1       C2       C3       C4       C5");
fprintf(fdata,"       C6       C7       C8       C9");
fprintf(fdata,"       C10      C11      C12\n");
for (k=0;k<samples;++k )
        {printf("Sample %2d = \n",k);
        fprintf(fdata,"Sample %2d = \n",k);
        for (j=0;j<=passthru;++j)
                {temp=eigenvalues[j];
                amount=(sqrt(temp)*eigenvectors[j][k]);
                printf("%8.3f ",amount);
                fprintf(fdata,"%8.3f ",amount);
                }
        printf("\n");
        fprintf(fdata,"\n");
        }
}         /*of function to display the loadings        */


get_irinfo(argv,hangar,size)/*  get from filename argv into array hangar[size]*/
char    *argv[];
float   *hangar;
int     size;           /*  size = limit to number of points to retrieve  */
{
FILE *f1;
float   temp;
int     i,j,k;

i=0;
f1=fopen(argv,"r+b");
while ((fscanf(f1,"%f",&temp)!=EOF)&&(i<size))
        {*(hangar+i)=temp/1000.;       /*amplitude mod to be removed later */
        ++i;
        }
fclose(f1);
}


needle_search()
{
int     i,j,k;

for (i=0;i<elements;++i)
        {printf("\n Search position = %d",i);
        design_testvector(i);
        predicted_from_testvector();
        compare_test_to_predicted(i);
        }
pick_best_and_store();
}


put_irinfo_into_data(matrix,col,info,no_elements)
float   matrix[RANGE][RANK],info[SPEC_SIZE];
int     no_elements,col;
{
int     i,j,k,l;

        start_wlength=(HIGH_F-where_to_begin)/DELTA_F;
        end_wlength=start_wlength+RANGE;
        elements=RANGE;
for (i=start_wlength,j=0;i<end_wlength;++i,++j)
        matrix[j][col]=info[i];
}

design_testvector(locale)
```

```
int     locale;
{
int     i,j,k;

for (i=0;i<elements;++i)
        testvector[i]=NOISE;
testvector[locale]=1.000;
}

predicted_from_testvector()
{
float   rhs[RANK];
int     i,j,k,l,m,n;

for (j=0;j<=passthru;++j)
        {rhs[j]=0.0;
        for (i=0;i<elements;++i)
                rhs[j]+=abst_row[i][j]*testvector[i];   /*abst row transpose*/
        rhs[j]=rhs[j]/eigenvalues[j];
        }
for (k=0;k<elements;++k)
        {predicted[k]=0.0;
        for (l=0;l<=passthru;++l)
                predicted[k]+=abst_row[k][l]*rhs[l];
        }
}

compare_test_to_predicted(position)
int     position;
{                       /* This is a NON-WEIGHTED risk analysis */
int     i,j,k;          /* This should be advanced at some point*/
                        /* For example in defining spectra one  */
                        /* should penalize negatives in predict.*/
float   temp,pure,weighted;

for (i=0,risk[position]=0.0,pure=0.0,weighted=0.0;i<elements;++i)
        {temp=(predicted[i]-testvector[i]);
        temp=temp*temp;
        pure+=temp;
        if (predicted[i]<0.0) weighted+=100*(predicted[i]*predicted[i]);
        else weighted+=predicted[i]*predicted[i];
        }
   risk[position]=pure*weighted;
}

pick_best_and_store()
{
FILE *fout;
float   minima[RANK],temp;
int     i,j,k,l,m;
char    fname[8];

fname[0]='X';
fname[2]=NULL;
for (i=0;i<=passthru;++i)
        {minima[i]=risk[i];     /* Have to start somewhere */
        best_guesses[i]=i;
        }
for (i=0;i<=passthru;++i)
        {for (j=i+1;j<elements;++j)
                {if (risk[j]<risk[best_guesses[i]])
                        {if (i==0)
                        best_guesses[i]=j;
                        else
                        if (risk[j]>risk[best_guesses[i-1]])
                                best_guesses[i]=j;
                        }
                }
        }
for (i=0;i<=passthru;++i)
        {fname[1]=65+i;
        fout=fopen(fname,"w");
        j=best_guesses[i];
        design_testvector(j);
        predicted_from_testvector();
        for (k=0,temp=where_to_begin+DELTA_F;k<elements;++k)
                {temp-=DELTA_F;
                fprintf(fout,"%8.2f %8.5f\n",temp,predicted[k]);
                }
        fclose(fout);
        }

/*  Now print out the risk function vs. frequency      */
fname[0]='R';fname[1]='I';fname[2]='S';fname[3]='K';fname[4]=NULL;
fout=fopen(fname,"w");
for (k=0,temp=where_to_begin+DELTA_F;k<elements;++k)
        {temp-=DELTA_F;
        fprintf(fout,"%8.2f %8.5f\n",temp,risk[k]);
        }
fclose(fout);

}
```

MODIFIED FACTOR ANALYSIS PROGRAM

```
/*      This program performs a factor analysis of              */
/*      all dominant factors in a random set of spectra         */

#include "math.h"
#include "stdio.h"
#include "stdlib.h"
#define SPEC_SIZE 3400  /* Total number of data points in spectra */
#define RANK 12         /* Typical number of eigenvalues expected */
#define RANGE 800       /* Typical number of wavelengths analyzed */
#define NOISE 0.08      /* 1000:1 T mode noise at 50% transmission*/
#define DELTA_F 1.0000  /* frequency spacing in wavenumbers       */
#define HIGH_F 4000     /* Starting frequency in cm-1 of Spectra  */


float   predicted[RANGE],testvector[RANGE],risk[RANGE];
float   irinfo[SPEC_SIZE],datamatrix[RANGE][RANK];
float   covariancematrix[RANK][RANK];
float   runningmatrix[RANK][RANK],subtrahend[RANK][RANK];
float   abst_row[RANGE][RANK],eigenvalues[RANK],guess[RANK],guess2[RANK];
float   crosselement,improvement,eigenvectors[RANK][RANK];
double  amplitude,oldamplitude,residual;
int     elements,samples,result,i,j,k,l,row,column,passthru,sampcnt;
int     start_wlength,end_wlength,where_to_begin,position,space_size;
char    query,query2,fname[1][14],dateinfo[30],inputname[RANK][20],fdataname[20];
FILE    *fdata;
int     best_guesses[RANK],lowpossible;
char    stopit;

main()
{
FILE    *f9;
float   temp;
int     sampcnt;

result=system("cls");
printf("          Please Enter the DATE (any format) ...");
scanf("%s",dateinfo);
printf("\n\n\n  *******      ABSTRACT FACTOR ANALYTICAL REDUCTION     ******\n");
printf("                 Analysis Date .... %s\n",dateinfo);
printf("          Enter file name for data output ......");
scanf("%s",fdataname);
fdata=fopen(fdataname,"w");
fprintf(fdata,"\n\n\n  *******      ABSTRACT FACTOR ANALYTICAL REDUCTION     ******\n");
fprintf(fdata,"                 Analysis Date .... %s\n",dateinfo);
printf("          Enter the number of samples ..........");
scanf("%d",&samples);
printf("          Will data be from IRD files? (y/n) ...");
query=getche();
passthru=-1;
```

```
if ((query!='y') && (query!='Y'))
        {query2='n';
        printf("\n          Enter the number of elements/sample  ");
        scanf("%d",&elements);
        }
if ((query=='y') || (query=='Y'))
        {lowpossible=440+(RANGE*1.0000);
        printf("\n          Enter --integer-- for starting frequency in cm-1 ... ");
        do
          {printf("\n          Value must be greater than %4d cm-1. Value = ",lowpossible);
          scanf("%d",&where_to_begin);
          }
        while (where_to_begin<lowpossible);
        printf("    Starting frequency in cm-1 = %5d \n",where_to_begin);
        printf("    %4d points used at a spacing of %6.4f cm-1",RANGE,DELTA_F);
        printf("\n");
        fprintf(fdata,"   Starting frequency in cm-1 = %5d \n",where_to_begin);
        fprintf(fdata,"   %4d points used at a spacing of %6.4f cm-1",RANGE,DELTA_F);
        fprintf(fdata,"\n");
        start_wlength=(HIGH_F-where_to_begin)/DELTA_F;
        end_wlength=start_wlength+RANGE;
        elements=RANGE;
        }
f9=fopen("BOOKDATA","r");
for (sampcnt=0;sampcnt<samples;++sampcnt )
        {printf("\n");
        if ((query=='y') || (query=='Y'))
                {printf("Name of file for sample # %3d = ",sampcnt);
                scanf("%s",fname);
                printf("\n");
                get_irinfo(fname,irinfo,SPEC_SIZE);
                put_irinfo_into_data(datamatrix,sampcnt,irinfo,elements);
                }
        else
                {for (j=0;j<elements;++j )
                        {printf("Element # %3d = ",j);
                        fscanf(f9,"%f",&temp);
                        datamatrix[j][sampcnt]=temp;
                        printf("%6.3f ",temp);
                        }
                }
        }
fclose(f9);
find_the_covariance_matrix();
printf("\n\n              ****** EIGENVALUES and EIGENVECTORS ******");
fprintf(fdata,"\n\n              ****** EIGENVALUES and EIGENVECTORS ******\n");
do
     {passthru=passthru+1;
     guess_the_initial_vector();
     run_x_initial_guess();
     load_vector_into_eigenvectors();
     find_new_running_matrix();
     get_residual();
     }
while ((passthru<samples-1) );
display_errors();
calc_abstract_row_matrix();
if ((query1!='y') && (query1!='Y') && (query2!='y') && (query2!='Y'))
     disp_abst_row_matrix();
find_loadings();
fclose(fdata);
/*needle_search();*/
}       /* End of procedure MAIN                          */



find_the_covariance_matrix()
{
int     i,k,l;

for (i=0;i<samples;++i )
        {for (k=0;k<samples;++k )
                {covariancematrix[i][k]=0.0;
                for (l=0;l<elements;++l)
                  {crosselement=datamatrix[l][i]*datamatrix[l][k];
                  covariancematrix[i][k]+=crosselement;
                  }
                runningmatrix[i][k]=covariancematrix[i][k];
                }
        }
}      /*of the function to find the covariance matrix DtD      */


guess_the_initial_vector()
{
double  temp;
int     i;

for (i=0;i<samples;++i )
        {temp=1.0/samples;
        guess[i]=sqrt(temp);
        }
}      /*of the function to guess the initial vector            */


run_x_initial_guess()
{
float   posneg;

amplitude=0.0;
```

```
do
     {for (i=0;i<samples;++i )
        {crosselement=0.0;
        for (j=0;j<samples;++j )
               crosselement+=runningmatrix[i][j]*guess[j];
        guess2[i]=crosselement;
        }
     oldamplitude=amplitude;
     amplitude=0.0;
     for (i=0;i<samples;++i )
          amplitude+=guess2[i]*guess2[i];
     for (i=0;i<samples;++i )
          guess[i]=guess2[i]/(sqrt(amplitude));
     printf("\n");
     posneg=(1.00-(oldamplitude/amplitude));
     if (posneg<0)posneg=-posneg;
     improvement=posneg;
     locatxy(15,20);printf("     ");locatxy(1,20);
     printf("improvement = %6.4f\n",improvement);
     }
while (improvement>0.001);
eigenvalues[passthru]=sqrt(amplitude);
printf("\n");
printf("Amplitude -Eigenvalue lambda- = %7.3f\n",eigenvalues[passthru]);
fprintf(fdata,"Amplitude -Eigenvalue lambda- = %7.3f\n",eigenvalues[passthru]);
printf("That eigenvector is  ....");
fprintf(fdata,"That eigenvector is  ....");
for (i=0;i<samples;++i )
        {printf("%6.3f   ",guess[i]);
        fprintf(fdata,"%6.3f   ",guess[i]);
        }
printf("\n");
fprintf(fdata,"\n");
}             /*of the function to mult. running matrix by guess vector */


get_residual()
{
float temp;

residual=0.0;
for (i=0;i<samples;++i )
   {for (j=0;j<samples;++j )
        {temp=runningmatrix[i][j];
        residual+=temp*temp;
        }
   }
}              /*of the function to find the residual sum of sqrs of matrix*/
```

```
find_new_running_matrix()
{
for (i=0;i<samples;++i )
        {for (j=0;j<samples;++j )
                subtrahend[i][j]=0.0;
        }
for (i=0;i<samples;++i )
        {for (j=0;j<samples;++j )
                {subtrahend[i][j]+=(eigenvalues[passthru]
                *eigenvectors[passthru][i]*eigenvectors[passthru][j]);
                }
        }
for (i=0;i<samples;++i)
        {for (j=0;j<samples;++j)
                runningmatrix[i][j]-=subtrahend[i][j];
        }
}            /*of function to find new running matrix   */


load_vector_into_eigenvectors()
{
for (i=0;i<samples;++i )
        eigenvectors[passthru][i]=guess[i];
}             /*of function to load iterated guess into eigenvectors    */


display_errors()
{
double  temp;
float   re,ie,xe,ind,ind_test;
int     u,t,n,j,infinity;

if (samples>elements)  u=samples; else u=elements;
if (samples <elements)  t=samples; else t=elements;
n=passthru;
space_size=1;   /*increment by one until indicator minimizes*/
printf("\n\n              *******  ERROR ANALYSIS  *******\n");
printf("RE = real error\n");printf("XE = extracted error\n");
printf("IE = imbedded error\n");
printf("IND= indicator function\n");
printf("\nEigenvalue    RE         XE          IE          IND\n");
printf("------------------------------------------------------\n");
fprintf(fdata,"\n\n              *******  ERROR ANALYSIS  *******\n");
fprintf(fdata,"RE = real error\n");printf("XE = extracted error\n");
fprintf(fdata,"IE = imbedded error\n");
fprintf(fdata,"IND= indicator function\n");
fprintf(fdata,"\nEigenvalue    RE         XE          IE          IND\n");
fprintf(fdata,"------------------------------------------------------\n");
for (n=0;n<=passthru;++n)
        {infinity=0;
        re=0;ie=0;xe=0;
        for (j=n+1;j<t;++j)
             {re+=eigenvalues[j];
             xe+=eigenvalues[j];
             }
        if (t>n)
             {temp=(re/(u*(t-n)));
             re=sqrt(temp);
             ind=re/((t-n)*(t-n));
             if (n==0)
                ind_test=ind;
             else if (ind<=ind_test)
                ++space_size;
             }
        if (t<=n) infinity=1 ;
        temp=xe/(u*t);
        xe=sqrt(temp);
        temp=(re*re)-(xe*xe);
        ie=sqrt(temp);
        if (infinity==0)
             {printf("%10.5f %10.5f %10.5f %10.5f %10.5f\n",
                eigenvalues[n],re,xe,ie,ind);
             fprintf(fdata,"%10.5f %10.5f %10.5f %10.5f %10.5f\n",
                eigenvalues[n],re,xe,ie,ind);
             }
        else {printf("%10.5f   ***    %10.5f   ***        ***\n",eigenvalues[n],xe);
              fprintf(fdata,"%10.5f   ***    %10.5f   ***        ***\n",eigenvalues[n],xe);
             }
        }
}           /*of function to find the errors and display them   */



calc_abstract_row_matrix()
{
float   storage;
int     i,j,k,l;

for (i=0;i<=passthru;++i)
        {for (j=0;j<elements;++j )
                {storage=0.0;
                for (k=0;k<samples;++k )
                       storage+=(datamatrix[j][k]*eigenvectors[i][k]);
                abst_row[j][i]=storage;
                }
        }
}           /* function to calculate abstract row matrix        */


disp_abst_row_matrix()
```

```
{
int     i,j,k,l;

printf("\n    Abstract Row Matrix R  ");
printf("\n-----------------------------\n");
fprintf(fdata,"\n    Abstract Row Matrix R  ");
fprintf(fdata,"\n----------------------------\n");
for (i=0;i<elements;++i)
        {for (j=0;j<=passthru;++j)
                {printf("%10.5f ",abst_row[i][j]);
                fprintf(fdata,"%10.5f ",abst_row[i][j]);
                }
        printf("\n");
        fprintf(fdata,"\n");
        }
printf("\n\n");
fprintf(fdata,"\n\n");
}         /*of function to display the abstract row matrix      */


find_loadings()
{
double  temp;
float   loading[RANK][RANK],amount;
int     i,j,k,l;

printf("\n\n                ",
       " ****** DATA COLUMN LOADINGS ONTO THE FACTORS ******");
printf("C=eigenvector (or factor) #\n");
printf("     C1       C2       C3       C4       C5");
printf("       C6       C7       C8       C9");
printf("       C10      C11      C12\n");
fprintf(fdata,"\n\n                ",
       " ****** DATA COLUMN LOADINGS ONTO THE FACTORS ******");
fprintf(fdata,"C=eigenvector (or factor) #\n");
fprintf(fdata,"     C1       C2       C3       C4       C5");
fprintf(fdata,"       C6       C7       C8       C9");
fprintf(fdata,"       C10      C11      C12\n");
for (k=0;k<samples;++k )
        {printf("Sample %2d = \n",k);
        fprintf(fdata,"Sample %2d = \n",k);
        for (j=0;j<=passthru;++j)
                {temp=eigenvalues[j];
                amount=(sqrt(temp)*eigenvectors[j][k]);
                printf("%8.3f ",amount);
                fprintf(fdata,"%8.3f ",amount);
                }
        printf("\n");
        fprintf(fdata,"\n");
        }
}          /*of function to display the loadings      */


get_irinfo(argv,hangar,size)/*  get from filename argv into array hangar[size]*/
char    *argv[];
float   *hangar;
int     size;           /*  size = limit to number of points to retrieve  */
{
FILE *f1;
float   temp;
int     i,j,k;

i=0;
f1=fopen(argv,"r+b");
while ((fscanf(f1,"%f",&temp)!=EOF)&&(i<size))
        {*(hangar+i)=temp;       /*amplitude mod to be removed later */
        ++i;
        }
fclose(f1);
}


needle_search()
{
int     i,j,k;

for (i=0;i<elements;++i)
        {printf("\n Search position = %d",i);
        design_testvector(i);
        predicted_from_testvector();
        compare_test_to_predicted(i);
        }
pick_best_and_store();
}


put_irinfo_into_data(matrix,col,info,no_elements)
float   matrix[RANGE][RANK],info[SPEC_SIZE];
int     no_elements,col;
{
int     i,j,k,l;

        start_wlength=(HIGH_F-where_to_begin)/DELTA_F;
        end_wlength=start_wlength+RANGE;
        elements=RANGE;
for (i=start_wlength,j=0;i<end_wlength;++i,++j)
        matrix[j][col]=info[i];
}

design_testvector(locale)
```

```
int     locale;
{
int     i,j,k;

for (i=0;i<elements;++i)
        testvector[i]=NOISE;
testvector[locale]=1.000;
}

predicted_from_testvector()
{
float   rhs[RANK];
int     i,j,k,l,m,n;

for (j=0;j<=passthru;++j)
        {rhs[j]=0.0;
        for (i=0;i<elements;++i)
                rhs[j]+=abst_row[i][j]*testvector[i];   /*abst row transpose*/
        rhs[j]=rhs[j]/eigenvalues[j];
        }
for (k=0;k<elements;++k)
        {predicted[k]=0.0;
        for (l=0;l<=passthru;++l)
                predicted[k]+=abst_row[k][l]*rhs[l];
        }
}

compare_test_to_predicted(position)
int     position;
{                       /* This is a NON-WEIGHTED risk analysis */
int     i,j,k;          /* This should be advanced at some point*/
                        /* For example in defining spectra one  */
                        /* should penalize negatives in predict.*/
float   temp,pure,weighted;

for (i=0,risk[position]=0.0,pure=0.0,weighted=0.0;i<elements;++i)
        {temp=(predicted[i]-testvector[i]);
        temp=temp*temp;
        pure+=temp;
        if (predicted[i]<0.0) weighted+=100*(predicted[i]*predicted[i]);
        else weighted+=predicted[i]*predicted[i];
        }
   risk[position]=pure*weighted;
}

pick_best_and_store()
{
FILE *fout;
float   minima[RANK],temp;
int     i,j,k,l,m;
char    fname[8];

fname[Ø]='X';
fname[2]=NULL;
for (i=Ø;i<=passthru;++i)
        {minima[i]=risk[i];     /* Have to start somewhere */
        best_guesses[i]=i;
        }
for (i=Ø;i<=passthru;++i)
        {for (j=i+1;j<elements;++j)
                {if (risk[j]<risk[best_guesses[i]])
                        {if (i==Ø)
                        best_guesses[i]=j;
                        else
                        if (risk[j]>risk[best_guesses[i-1]])
                                best_guesses[i]=j;
                        }
                }
        }
for (i=Ø;i<=passthru;++i)
        {fname[1]=65+i;
        fout=fopen(fname,"w");
        j=best_guesses[i];
        design_testvector(j);
        predicted_from_testvector();
        for (k=Ø,temp=where_to_begin+DELTA_F;k<elements;++k)
                {temp-=DELTA_F;
                fprintf(fout,"%8.2f %8.5f\n",temp,predicted[k]);
                }
        fclose(fout);
        }

/*   Now print out the risk function vs. frequency       */
fname[Ø]='R';fname[1]='I';fname[2]='S';fname[3]='K';fname[4]=NULL;
fout=fopen(fname,"w");
for (k=Ø,temp=where_to_begin+DELTA_F;k<elements;++k)
        {temp-=DELTA_F;
        fprintf(fout,"%8.2f %8.5f\n",temp,risk[k]);
        }
fclose(fout);

}
```

PROGRAM TO TRANSFER SPECTRAL DATA FROM SPECTROMETER TO THE EXTERNAL COMPUTER

```
/*    This program allows the user to obtain the spectral data points        */
/*    of a spectrum scanned into the memory areas of the Perkin Elmer        */
/*    1700 FT - IR spectrometer                                              */

#include <stdio.h>
#include <conio.h>
#include <fcntl.h>
#include <float.h>
#include <io.h>
#include <math.h>
#include <stdlib.h>
#include <types.h>
#include <process.h>

unsigned char spect_data[5000];    /* Array in which spectral data is stored */
unsigned char spect_temp[20];      /* Array used for data conversion         */
char          fdataname[20];

int      i,j,k,l,m,n,p,nc,ch;
int      memory,high,inter,low,numread,result;
float    absorb,magnitude,trans;
FILE     *fdata;

/*   Where                                                                   */
/*               memory =  memory area to obtain spectrum from               */
/*                         reference area = 0 ; sample area = 1              */
/*               high   =  starting frequency of spectrum in cm-1            */
/*               low    =  ending frequency of spectrum in cm-               */
/*               inter  =  data interval of spectrum                         */
/*               absorb =  spectral data as absorbance                       */
/*               trans  =  spectral data as transmittance                    */

main()

{


FILE     *pe;
FILE     *sk;



result=system("CLS");

printf("         ******WELCOME TO THE SPECTRUM GRABBER !!!!****** \n\n\n");
printf(" Data Acquisition and Control Software, Version 1.0  March 25th 1988\n\n\n");

start:

printf(" Please enter ‘filename.ird’ for data output\n ");
```

```
scanf("%s",fdataname);
printf("\n\n");

printf(" Please enter memory area to grab spectrum from\n");
printf(" Allowed memory areas are 1 = A, 2 = B, 3 = C, 4 = Difference.\n\n");

scanf("%d",&memory);

        while((memory < 1) || (memory > 4))
                {
                printf(" Wrong memory area, please reenter!\n");
                scanf("%d",&memory);
                }

printf("\n\n");

printf(" Please enter the starting wavenumber for spectral data \n ");
printf(" Starting wavenumber must be between 4400 and 402 cm-1\n");

scanf("%d",&high);

        while((high < 402) || (high > 4400))
                {
                printf(" Incorrect starting wavenumber!! Please reenter!\n");
                scanf("%d",&high);
                }

printf("\n\n");

printf(" Now enter the ending wavenumber for spectral data \n");
printf(" Ending wavenumber must be between 400 and 4398 cm-1\n");

scanf("%d",&low);

        while((low > 4398) || (low < 400))
                {
                printf(" Incorrect ending wavenumber!! Please reenter!\n");
                scanf("%d",&low);
                }

printf("\n\n");

printf(" Finally, Please enter the data interval for the spectrum \n");
printf(" Data interval must be an integer value * 100 !!!\n");

scanf("%d",&inter);

        while((inter < 100) || (inter > 1000))
                {
                printf(" Incorrect data interval!!! Please reenter\n");
                scanf("%d",&inter);
                }

printf("\n\n");

printf(" I am busy, please wait ......\n");
printf("\n\n");

/*   Open COM1 to send and receive data                                       */

        pe=fopen("COM1","w");
        sk=fopen("COM1","r");

/*   Open the file to write the spectral data                                 */

        fdata=fopen(fdataname,"w");

/*   Reset the PE 1700 to ASCI mode                                           */

        result=fprintf(pe,"$ASCI\r");
        fflush(pe);

/*   Read the port to obtain return value                                     */

                numread=fread(spect_data,1,4,sk);
                for(i=3;i<numread;++i);
                printf("%c",spect_data[i]);
                printf("\n");
                fflush(sk);

/*  Send the MOVE command to the spectrometer                                 */

        result=fprintf(pe,"$MOVE %d %d %d %d\r",high,low,inter,memory);
        fflush(pe);

/*   Read the port to obtain return value                                     */

                numread=fread(spect_data,1,10,sk);
                for(i=0;i<numread;++i)
                printf("%c",spect_data[i]);
                printf("\n");
                fflush(sk);

/*  Send the MOVE command to the spectrometer again to initiate spectral      */
/*  transfer.  This procedure is necessary to obtain the data.                */

        while (spect_data[2]!='6')

                {
                result=fprintf(pe,"$MOVE\r");
        fflush(pe);

        numread=fread(spect_data,1,4004,sk);
        for(i=4;i<numread;)
                {
                trans=0.0;
                magnitude=0.01;
                nc=0;
                p=i;

                do
                        {
                          if((spect_data[p] >= 48) && (spect_data[p] <= 57))
                                {
                                  spect_temp[nc] = spect_data[p];
                                  ++nc;
                                }

                                  ++p;
                        }

        while((spect_data[p-1] >= 48) && (spect_data[p-1] <= 57));

                  for(m=nc-1;m>=0;--m)
                        {
                          trans+=magnitude*(spect_temp[m]-48.0);
                          magnitude*=10.0;
                        }

                          trans*=0.5;
                          absorb=2.0-log10(trans);
                          fprintf(fdata,"%8.4f",absorb);
                          i=p;

                }

        }


fflush(sk);

result=fprintf(pe,"$RLSE\r");
fflush(pe);


printf("\n\n");
printf(" Do you wish to obtain another spectrum ? (y/n)\n\n");

ch = getche();
```

```
                if((ch == 'y') || (ch == 'Y'))
                        {printf("\n\n\n");
                        goto start;}
                else
                        printf("\n\n");
                        printf(" Thank you.  Bye !!\n");


        result=fprintf(pe,"$RLSE\r");
        fflush(pe);

fclose(pe);
fclose(sk);

}
```

APPENDIX B

RESULTS FROM FACTOR ANALYSIS

```
*******     ABSTRACT FACTOR ANALYTICAL REDUCTION    ******
                 Analysis Date .... 5-7-88
 Starting frequency in cm-1 =  1400
  800 points used at a spacing of 1.0000 cm-1


              ****** EIGENVALUES and EIGENVECTORS ******
Amplitude -Eigenvalue lambda- =  36.817
That eigenvector is  .... 0.257   0.623   0.273   0.251   0.341   0.308   0.326   0.258   0.137   0.062
Amplitude -Eigenvalue lambda- =   1.242
That eigenvector is  ....-0.390  -0.242   0.263   0.429   0.507  -0.382  -0.068   0.187  -0.185   0.243
Amplitude -Eigenvalue lambda- =   0.384
That eigenvector is  .... 0.583  -0.585  -0.078   0.225  -0.043   0.174   0.012   0.459   0.089   0.099
Amplitude -Eigenvalue lambda- =   0.083
That eigenvector is  .... 0.109  -0.097  -0.107  -0.077   0.399   0.097  -0.235  -0.452   0.612   0.398
Amplitude -Eigenvalue lambda- =   0.037
That eigenvector is  ....-0.438  -0.203   0.100   0.125  -0.351   0.474   0.430  -0.074   0.036   0.449
Amplitude -Eigenvalue lambda- =   0.026
That eigenvector is  ....-0.416  -0.124   0.100  -0.008   0.017   0.130  -0.080   0.343   0.611  -0.536
Amplitude -Eigenvalue lambda- =   0.012
That eigenvector is  .... 0.075   0.196  -0.190   0.409  -0.423  -0.589   0.222  -0.007   0.401   0.129
Amplitude -Eigenvalue lambda- =   0.005
That eigenvector is  ....-0.008   0.163   0.400  -0.441  -0.259  -0.151  -0.357   0.424   0.113   0.458
Amplitude -Eigenvalue lambda- =   0.002
That eigenvector is  .... 0.219  -0.276   0.632  -0.261   0.026  -0.260   0.463  -0.287   0.103  -0.182
Amplitude -Eigenvalue lambda- =   0.002
That eigenvector is  .... 0.108   0.036   0.467   0.502  -0.310   0.204  -0.509  -0.315  -0.036  -0.139


              *******  ERROR ANALYSIS  *******
RE = real error
IE = imbedded error
IND= indicator function

Eigenvalue    RE         XE          IE          IND
-------------------------------------------------------
  36.81734    0.01497    0.01497    0.00000    0.00015
   1.24248    0.00875    0.00830    0.00277    0.00011
   0.38398    0.00511    0.00457    0.00228    0.00008
   0.08254    0.00388    0.00325    0.00212    0.00008
   0.03728    0.00313    0.00242    0.00198    0.00009
   0.02639    0.00227    0.00161    0.00161    0.00009
   0.01200    0.00164    0.00104    0.00127    0.00010
   0.00495    0.00124    0.00068    0.00103    0.00014
   0.00215    0.00097    0.00044    0.00087    0.00024
   0.00152    0.00000    0.00000    0.00000    0.00000
```

```
              C=eigenvector (or factor) #
   C1        C2        C3        C4        C5        C6        C7        C8        C9        C1Ø       C11       C12
Sample  Ø =
   1.562   -Ø.434    Ø.361    Ø.Ø31    -Ø.Ø85   -Ø.Ø68    Ø.ØØ8    -Ø.ØØ1    Ø.Ø1Ø    Ø.ØØ4
Sample  1 =
   3.781   -Ø.27Ø   -Ø.363   -Ø.Ø28    -Ø.Ø39   -Ø.Ø2Ø    Ø.Ø21    Ø.Ø11    -Ø.Ø13    Ø.ØØ1
Sample  2 =
   1.66Ø    Ø.293   -Ø.Ø48   -Ø.Ø31    Ø.Ø19    Ø.Ø16    -Ø.Ø21    Ø.Ø28    Ø.Ø29    Ø.Ø18
Sample  3 =
   1.526    Ø.478    Ø.14Ø   -Ø.Ø22    Ø.Ø24    -Ø.ØØ1    Ø.Ø45    -Ø.Ø31   -Ø.Ø12    Ø.Ø2Ø
Sample  4 =
   2.Ø72    Ø.566   -Ø.Ø27    Ø.115    -Ø.Ø68    Ø.ØØ3    -Ø.Ø46   -Ø.Ø18    Ø.ØØ1    -Ø.Ø12
Sample  5 =
   1.869   -Ø.426    Ø.1Ø8    Ø.Ø28    Ø.Ø92     Ø.Ø21    -Ø.[illegible]   -Ø.Ø11   -Ø.Ø12    Ø.ØØ8
Sample  6 =
   1.98Ø   -Ø.Ø75    Ø.ØØ7   -Ø.Ø67    Ø.Ø83    -Ø.Ø13    Ø.Ø24    -Ø.Ø25    Ø.Ø21    -Ø.Ø2Ø
Sample  7 =
   1.568    Ø.2Ø9    Ø.284   -Ø.13Ø    -Ø.Ø14    Ø.Ø56    -Ø.ØØ1    Ø.Ø3Ø    -Ø.Ø13   -Ø.Ø12
Sample  8 =
   Ø.831   -Ø.2Ø7    Ø.Ø55    Ø.176    Ø.ØØ7     Ø.Ø99    Ø.Ø44     Ø.ØØ8    Ø.ØØ5    -Ø.ØØ1
Sample  9 =
   Ø.378    Ø.271    Ø.Ø61    Ø.114    Ø.Ø87    -Ø.Ø87    Ø.Ø14     Ø.Ø32    -Ø.ØØ8   -Ø.ØØ5
```

```
 *******      ABSTRACT FACTOR ANALYTICAL REDUCTION    ******
                 Analysis Date .... 5-9-88
  Starting frequency in cm-1 =  2000
   800 points used at a spacing of 1.0000 cm-1


              ****** EIGENVALUES and EIGENVECTORS ******
Amplitude -Eigenvalue lambda- =  13.602
That eigenvector is  ....-0.024   0.415    0.386    0.434    0.541    0.081    0.240    0.347   -0.016    0.111
Amplitude -Eigenvalue lambda- =  0.251
That eigenvector is  .... 0.177  -0.096   -0.144   -0.012   -0.032    0.298    0.250    0.047    0.826    0.320
Amplitude -Eigenvalue lambda- =  0.080
That eigenvector is  .... 0.278  -0.118   -0.172   -0.001   -0.169    0.338    0.422    0.061   -0.558    0.494
Amplitude -Eigenvalue lambda- =  0.013
That eigenvector is  .... 0.492   0.325   -0.097   -0.248    0.080    0.540   -0.169    0.086   -0.042   -0.495
Amplitude -Eigenvalue lambda- =  0.005
That eigenvector is  .... 0.448   0.263    0.366    0.144   -0.604   -0.379    0.214    0.048    0.085   -0.101
Amplitude -Eigenvalue lambda- =  0.001
That eigenvector is  .... 0.496   0.184   -0.451   -0.006    0.378   -0.510   -0.191   -0.096   -0.002    0.259
Amplitude -Eigenvalue lambda- =  0.000
That eigenvector is  .... 0.261  -0.481    0.171    0.211   -0.063    0.045   -0.546    0.554   -0.014    0.127
Amplitude -Eigenvalue lambda- =  0.000
That eigenvector is  .... 0.120   0.083    0.604   -0.409    0.125    0.105   -0.283   -0.404   -0.004    0.420
Amplitude -Eigenvalue lambda- =  0.000
That eigenvector is  .... 0.338  -0.584    0.216    0.166    0.342   -0.052    0.322   -0.357   -0.011   -0.346
Amplitude -Eigenvalue lambda- =  0.000
That eigenvector is  ....-0.021  -0.106    0.125   -0.678    0.198   -0.275    0.337    0.530    0.004   -0.066


              *******  ERROR ANALYSIS  *******
RE = real error
IE = imbedded error
IND= indicator function

Eigenvalue    RE         XE         IE         IND
--------------------------------------------------------
  13.60209    0.00662    0.00662    0.00000    0.00007
   0.25117    0.00371    0.00352    0.00117    0.00005
   0.08027    0.00172    0.00154    0.00077    0.00003
   0.01281    0.00105    0.00088    0.00057    0.00002
   0.00460    0.00057    0.00044    0.00036    0.00002
   0.00128    0.00026    0.00018    0.00018    0.00001
   0.00013    0.00020    0.00013    0.00016    0.00001
   0.00010    0.00011    0.00006    0.00009    0.00001
   0.00002    0.00007    0.00003    0.00006    0.00002
   0.00001    0.00000    0.00000    0.00000    0.00000
```

```
                C=eigenvector (or factor) #
    C1        C2        C3        C4        C5        C6        C7        C8        C9        C10       C11       C12
Sample  0 =
  -0.090    0.089     0.079     0.056     0.030     0.018     0.003     0.001     0.002    -0.000
Sample  1 =
   1.531   -0.048    -0.033     0.037     0.018     0.007    -0.006     0.001    -0.003    -0.000
Sample  2 =
   1.425   -0.072    -0.049    -0.011     0.025    -0.016     0.002     0.006     0.001     0.000
Sample  3 =
   1.600   -0.006    -0.000    -0.028     0.010    -0.000     0.002    -0.004     0.001    -0.002
Sample  4 =
   1.994   -0.016    -0.048     0.009    -0.041     0.014    -0.001     0.001     0.002     0.001
Sample  5 =
   0.300    0.149     0.096     0.061    -0.026    -0.018     0.001     0.001    -0.000    -0.001
Sample  6 =
   0.885    0.125     0.120    -0.019     0.015    -0.007    -0.006    -0.003     0.001     0.001
Sample  7 =
   1.280    0.024     0.017     0.010     0.003    -0.003     0.006    -0.004    -0.002     0.001
Sample  8 =
  -0.060    0.414    -0.158    -0.005     0.006    -0.000    -0.000    -0.000    -0.000     0.000
Sample  9 =
   0.409    0.160     0.140    -0.056    -0.007     0.009     0.001     0.004    -0.002    -0.000
```

```
*******     ABSTRACT FACTOR ANALYTICAL REDUCTION    ******
                 Analysis Date .... 5-9-88
  Starting frequency in cm-1 =  3000
   800 points used at a spacing of 1.0000 cm-1


              ****** EIGENVALUES and EIGENVECTORS ******
Amplitude -Eigenvalue lambda- =  34.411
That eigenvector is  .... 0.028   0.394    0.362    0.409    0.516    0.198    0.274    0.352    0.092    0.182
Amplitude -Eigenvalue lambda- =   0.121
That eigenvector is  .... 0.640   0.013    0.444   -0.239   -0.335    0.016   -0.096    0.140    0.426    0.115
Amplitude -Eigenvalue lambda- =   0.022
That eigenvector is  ....-0.033  -0.138   -0.369   -0.040   -0.068    0.752    0.087   -0.099    0.264    0.430
Amplitude -Eigenvalue lambda- =   0.001
That eigenvector is  .... 0.105  -0.136    0.012    0.072   -0.204   -0.228    0.886   -0.267   -0.023    0.119
Amplitude -Eigenvalue lambda- =   0.000
That eigenvector is  ....-0.076   0.043   -0.443    0.270   -0.021   -0.523   -0.107    0.206    0.559    0.290
Amplitude -Eigenvalue lambda- =   0.000
That eigenvector is  .... 0.492   0.539   -0.284    0.263   -0.033   -0.015   -0.165   -0.472   -0.243    0.085
Amplitude -Eigenvalue lambda- =   0.000
That eigenvector is  ....-0.382   0.333    0.261   -0.018    0.018    0.069    0.005   -0.540    0.548   -0.279
Amplitude -Eigenvalue lambda- =   0.000
That eigenvector is  .... 0.198  -0.049   -0.039   -0.392    0.701   -0.149   -0.024   -0.486    0.221    0.058
Amplitude -Eigenvalue lambda- =   0.000
That eigenvector is  ....-0.197   0.050    0.039    0.392   -0.701    0.150    0.024    0.486   -0.221   -0.059
Amplitude -Eigenvalue lambda- =   0.000
That eigenvector is  ....-0.197   0.050    0.039    0.392   -0.701    0.150    0.024    0.486   -0.221   -0.059


              *******  ERROR ANALYSIS  *******
RE = real error
IE = imbedded error
IND= indicator function

Eigenvalue    RE         XE         IE         IND
-------------------------------------------------------
  34.41097    0.00424    0.00424    0.00000    0.00004
   0.12073    0.00180    0.00171    0.00057    0.00002
   0.02172    0.00052    0.00046    0.00023    0.00001
   0.00131    0.00026    0.00022    0.00014    0.00001
   0.00028    0.00015    0.00012    0.00010    0.00000
   0.00001    0.00016    0.00011    0.00011    0.00001
   0.00001    0.00017    0.00010    0.00013    0.00001
   0.00001    0.00018    0.00010    0.00015    0.00002
   0.00003    0.00018    0.00008    0.00016    0.00004
   0.00005    0.00000    0.00000    0.00000    0.00000
```

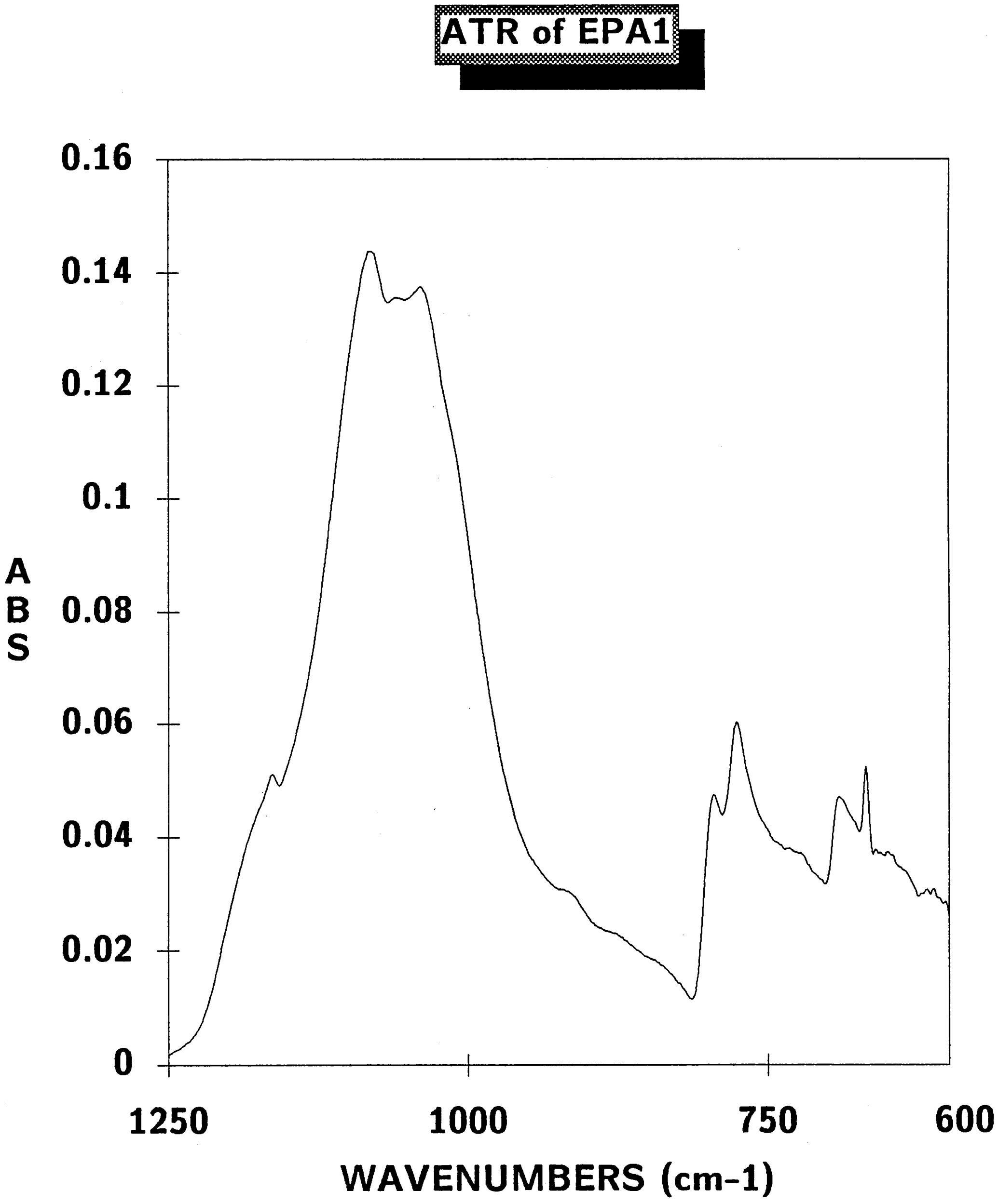
```
               C=eigenvector (or factor) #
     C1        C2        C3        C4        C5        C6        C7        C8        C9        C10       C11       C12
Sample  0 =
   0.164     0.223    -0.005     0.004    -0.001     0.002    -0.001     0.001    -0.001    -0.001
Sample  1 =
   2.310     0.005    -0.020    -0.005     0.001     0.002     0.001    -0.000     0.000     0.000
Sample  2 =
   2.124     0.154    -0.054     0.000    -0.007    -0.001     0.001    -0.000     0.000     0.000
Sample  3 =
   2.397    -0.083    -0.006     0.003     0.004     0.001    -0.000    -0.001     0.002     0.003
Sample  4 =
   3.028    -0.116    -0.010    -0.007    -0.000    -0.000     0.000     0.002    -0.004    -0.005
Sample  5 =
   1.160     0.005     0.111    -0.008    -0.009    -0.000     0.000    -0.001     0.001     0.001
Sample  6 =
   1.610    -0.033     0.013     0.032    -0.002    -0.001     0.000    -0.000     0.000     0.000
Sample  7 =
   2.063     0.049    -0.015    -0.010     0.003    -0.002    -0.002    -0.002     0.002     0.003
Sample  8 =
   0.542     0.148     0.039    -0.001     0.009    -0.001     0.002     0.001    -0.001    -0.002
Sample  9 =
   1.066     0.040     0.063     0.004     0.005     0.000    -0.001     0.000    -0.000    -0.000
```

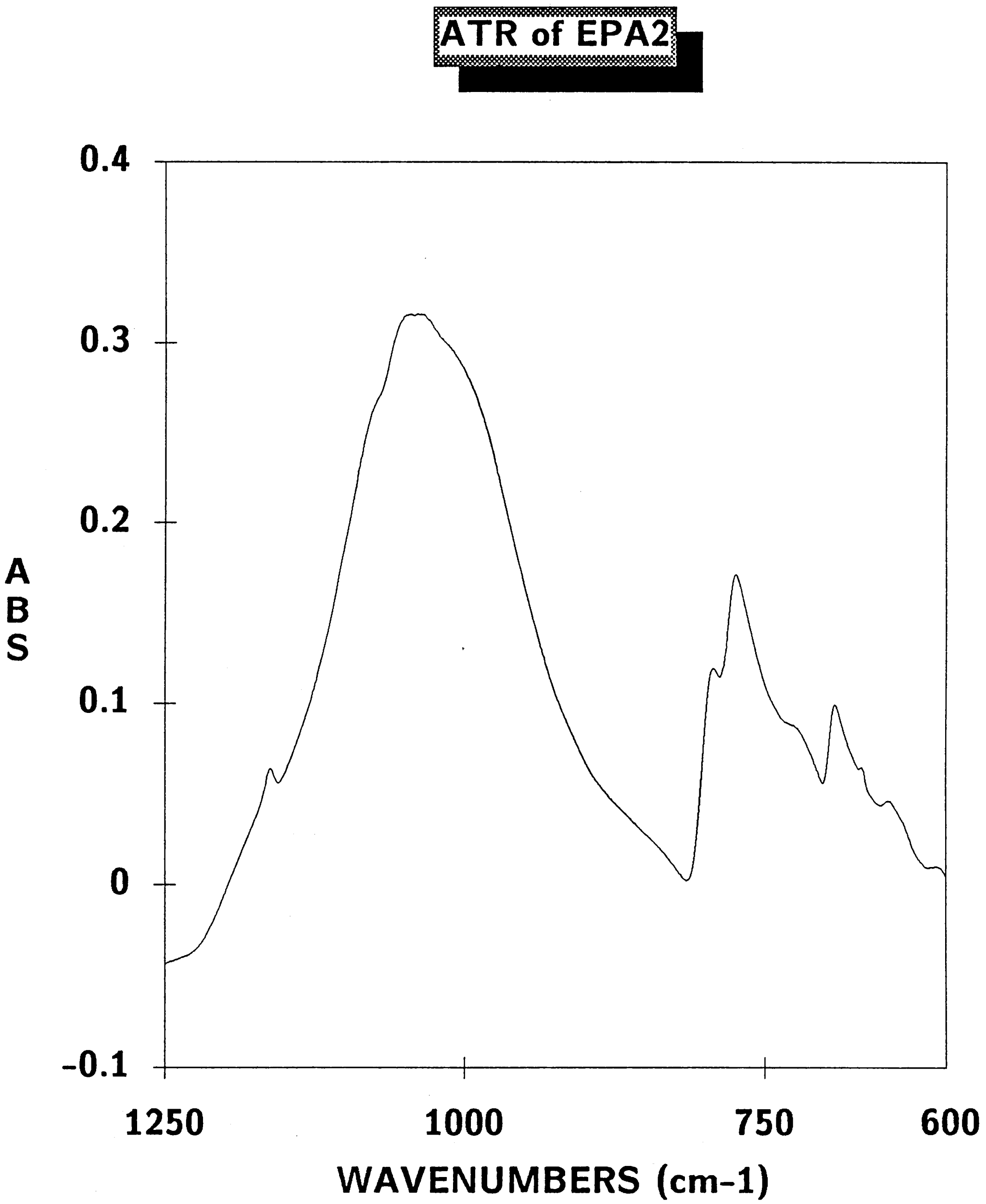
```
*******      ABSTRACT FACTOR ANALYTICAL REDUCTION     ******
                 Analysis Date .... 5-10-88
  Starting frequency in cm-1 =  4000
   800 points used at a spacing of 1.0000 cm-1


               ****** EIGENVALUES and EIGENVECTORS ******
Amplitude -Eigenvalue lambda- =  18.036
That eigenvector is  ....-0.060   0.356   0.411   0.393   0.531   0.225   0.192   0.373   0.076   0.178
Amplitude -Eigenvalue lambda- =  0.136
That eigenvector is  .... 0.534   0.143  -0.190  -0.052  -0.233   0.251   0.385  -0.088   0.478   0.393
Amplitude -Eigenvalue lambda- =  0.027
That eigenvector is  .... 0.158   0.471   0.404   0.105  -0.588   0.129   0.056  -0.012  -0.219  -0.405
Amplitude -Eigenvalue lambda- =  0.008
That eigenvector is  .... 0.375   0.481  -0.052  -0.088   0.330  -0.644  -0.163  -0.208   0.099  -0.119
Amplitude -Eigenvalue lambda- =  0.002
That eigenvector is  .... 0.218  -0.103  -0.509   0.381  -0.067  -0.166   0.174   0.583  -0.053  -0.363
Amplitude -Eigenvalue lambda- =  0.001
That eigenvector is  .... 0.410  -0.294   0.366  -0.026  -0.129  -0.044  -0.621   0.388   0.170   0.162
Amplitude -Eigenvalue lambda- =  0.001
That eigenvector is  .... 0.073  -0.429   0.460   0.046  -0.075  -0.510   0.571  -0.030  -0.038   0.047
Amplitude -Eigenvalue lambda- =  0.000
That eigenvector is  .... 0.410  -0.299   0.116  -0.122   0.414   0.399   0.078  -0.238   0.015  -0.567
Amplitude -Eigenvalue lambda- =  0.000
That eigenvector is  .... 0.372  -0.066  -0.100   0.343   0.058   0.072  -0.063  -0.245  -0.715   0.381
Amplitude -Eigenvalue lambda- =  0.000
That eigenvector is  ....-0.121  -0.132   0.022   0.741  -0.091  -0.045  -0.200  -0.435   0.410  -0.105


               *******  ERROR ANALYSIS  *******
RE = real error
IE = imbedded error
IND= indicator function

Eigenvalue     RE         XE          IE          IND
-------------------------------------------------------
  18.03569   0.00469    0.00469    0.00000    0.00005
   0.13647   0.00234    0.00222    0.00074    0.00003
   0.02716   0.00138    0.00123    0.00062    0.00002
   0.00793   0.00087    0.00073    0.00048    0.00002
   0.00208   0.00067    0.00052    0.00043    0.00002
   0.00118   0.00050    0.00035    0.00035    0.00002
   0.00069   0.00031    0.00020    0.00024    0.00002
   0.00023   0.00018    0.00010    0.00015    0.00002
   0.00005   0.00012    0.00005    0.00010    0.00003
   0.00002   0.00000    0.00000    0.00000    0.00000
```

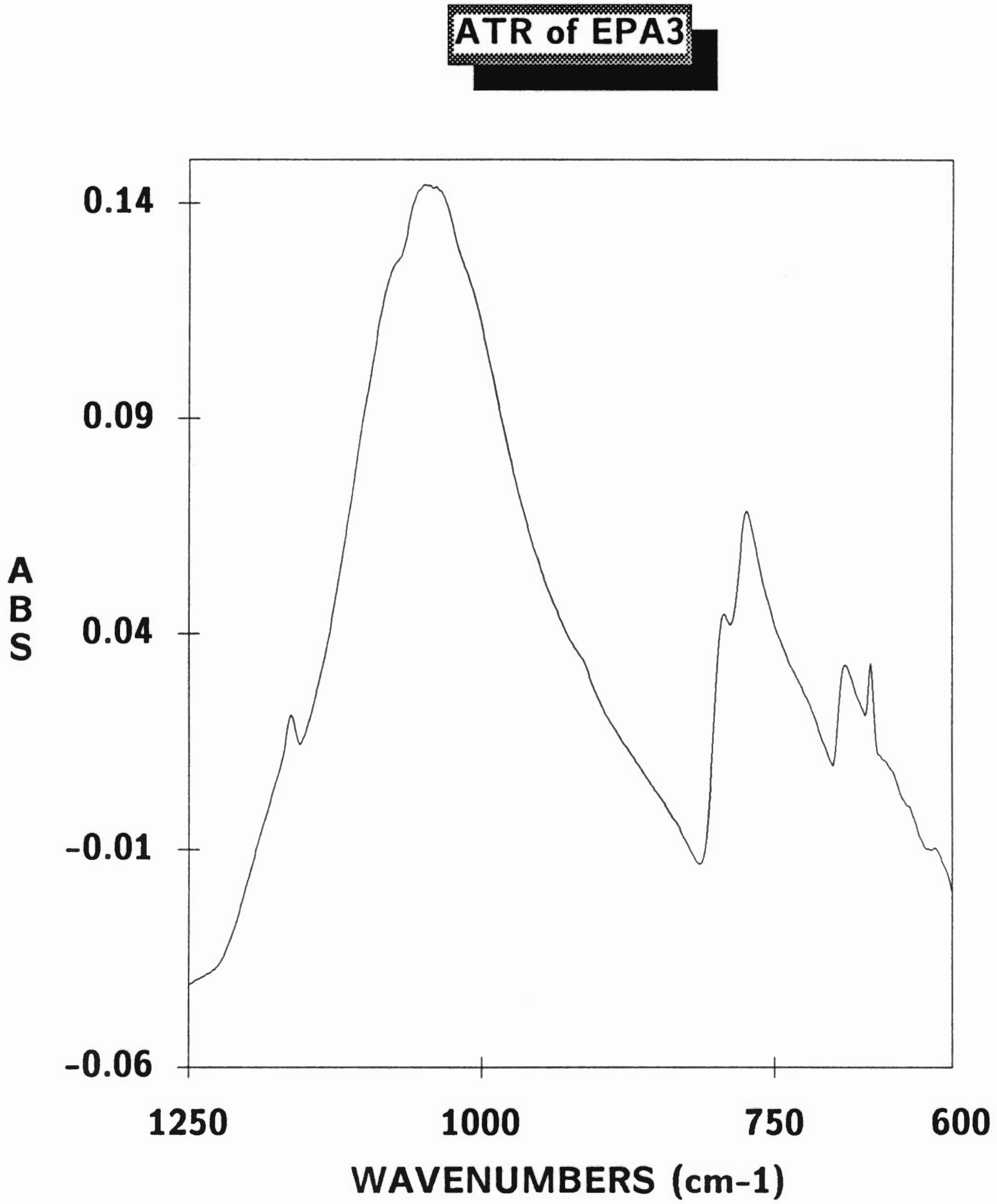
```
              C=eigenvector (or factor) #
     C1       C2       C3       C4       C5       C6       C7       C8       C9       C10      C11      C12
Sample  0 =
  -0.256    0.197    0.026    0.033    0.010    0.014    0.002    0.006    0.003   -0.001
Sample  1 =
   1.510    0.053    0.078    0.043   -0.005   -0.010   -0.011   -0.005   -0.000   -0.001
Sample  2 =
   1.747   -0.070    0.067   -0.005   -0.023    0.013    0.012    0.002   -0.001    0.000
Sample  3 =
   1.668   -0.019    0.017   -0.008    0.017   -0.001    0.001   -0.002    0.003    0.003
Sample  4 =
   2.254   -0.086   -0.097    0.029   -0.003   -0.004   -0.002    0.006    0.000   -0.000
Sample  5 =
   0.958    0.093    0.021   -0.057   -0.008   -0.002   -0.013    0.006    0.001   -0.000
Sample  6 =
   0.815    0.142    0.009   -0.015    0.008   -0.021    0.015    0.001   -0.000   -0.001
Sample  7 =
   1.586   -0.032   -0.002   -0.018    0.027    0.013   -0.001   -0.004   -0.002   -0.002
Sample  8 =
   0.321    0.177   -0.036    0.009   -0.002    0.006   -0.001    0.000   -0.005    0.002
Sample  9 =
   0.756    0.145   -0.067   -0.011   -0.017    0.006    0.001   -0.009    0.003   -0.000
```

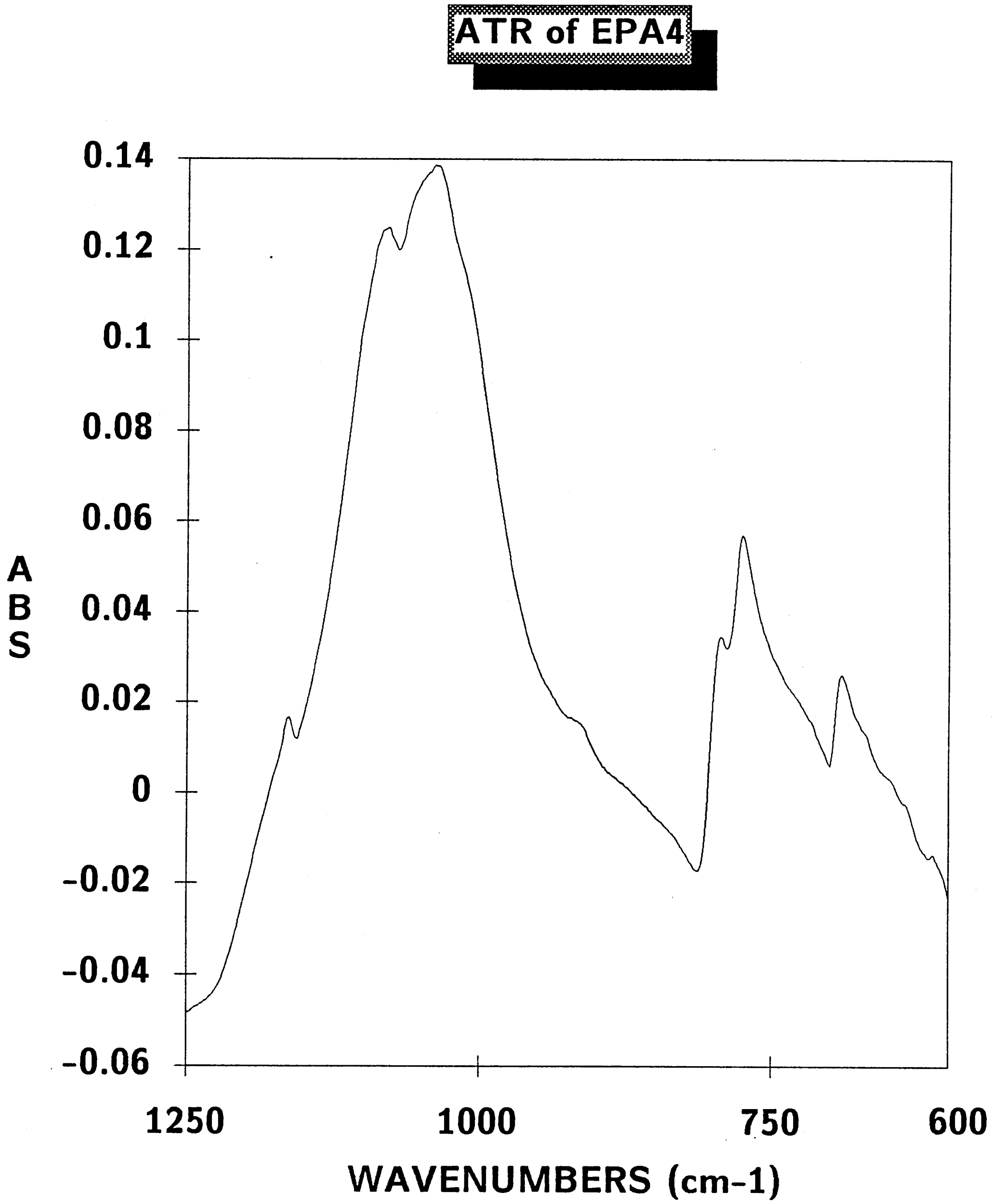
APPENDIX C

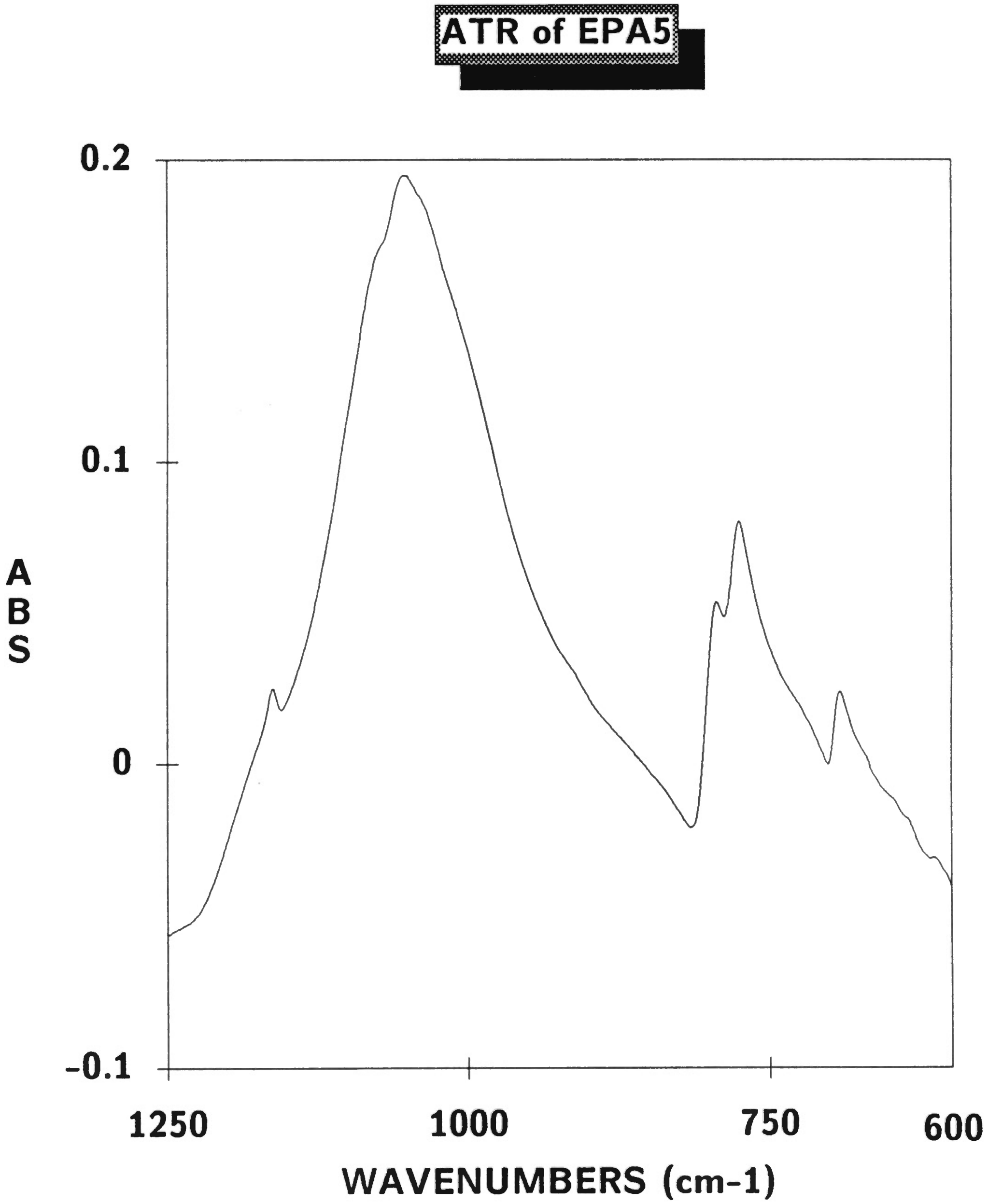
FTIR - ATR SPECTRA OF EPA1 - EPA9

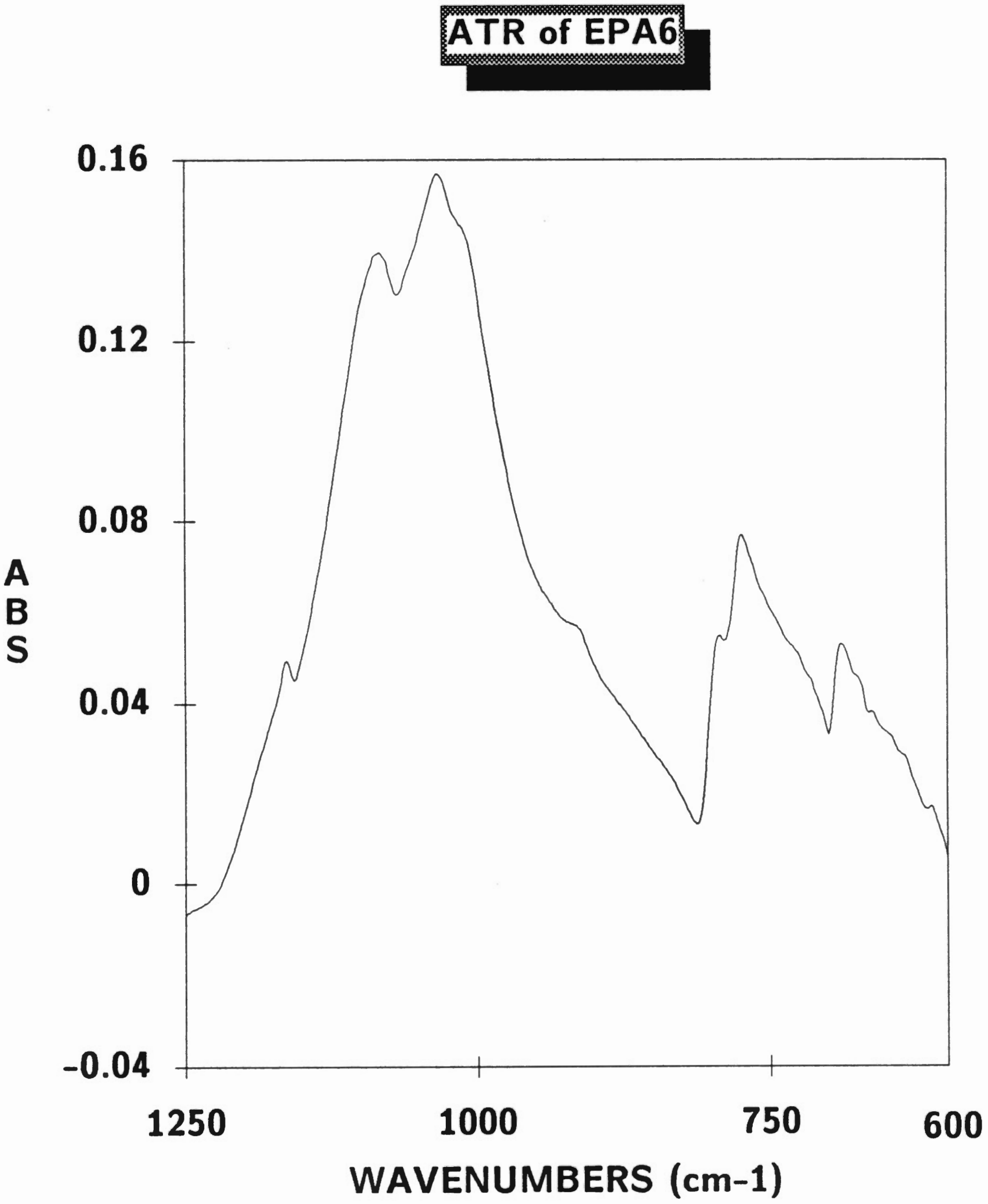

ATR of EPA1
0.16
0.14
0.12
0.1
A
B
S
0.08
0.06
0.04
0.02
0
1250
1000
750
600
WAVENUMBERS (cm-1)

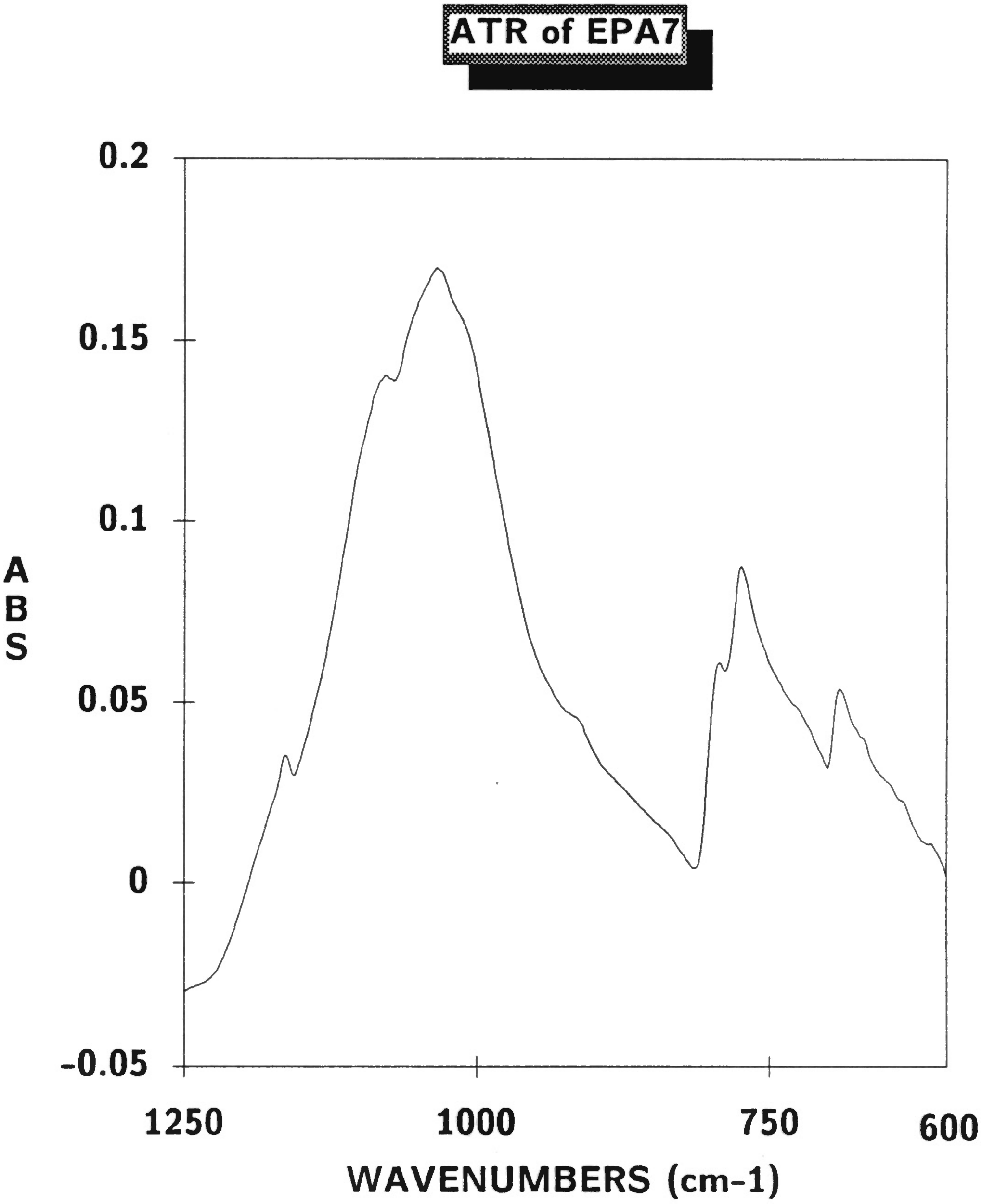

ATR of EPA2
0.4
0.3
0.2
0.1
0
-0.1
A
B
S
1250
1000
750
600
WAVENUMBERS (cm-1)

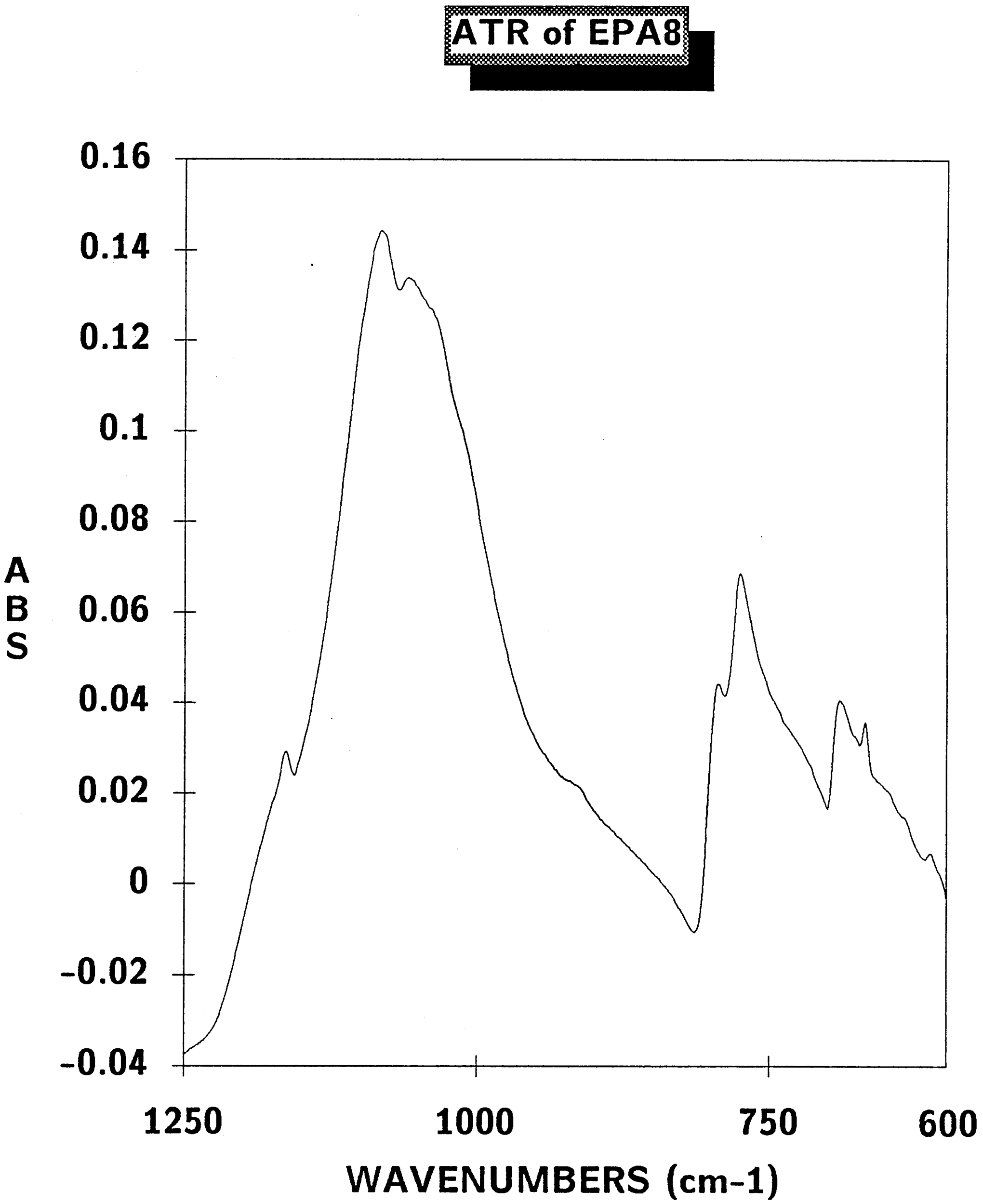

ATR of EPA3
A
B
S
0.14
0.09
0.04
-0.01
-0.06
1250
1000
750
600
WAVENUMBERS (cm-1)

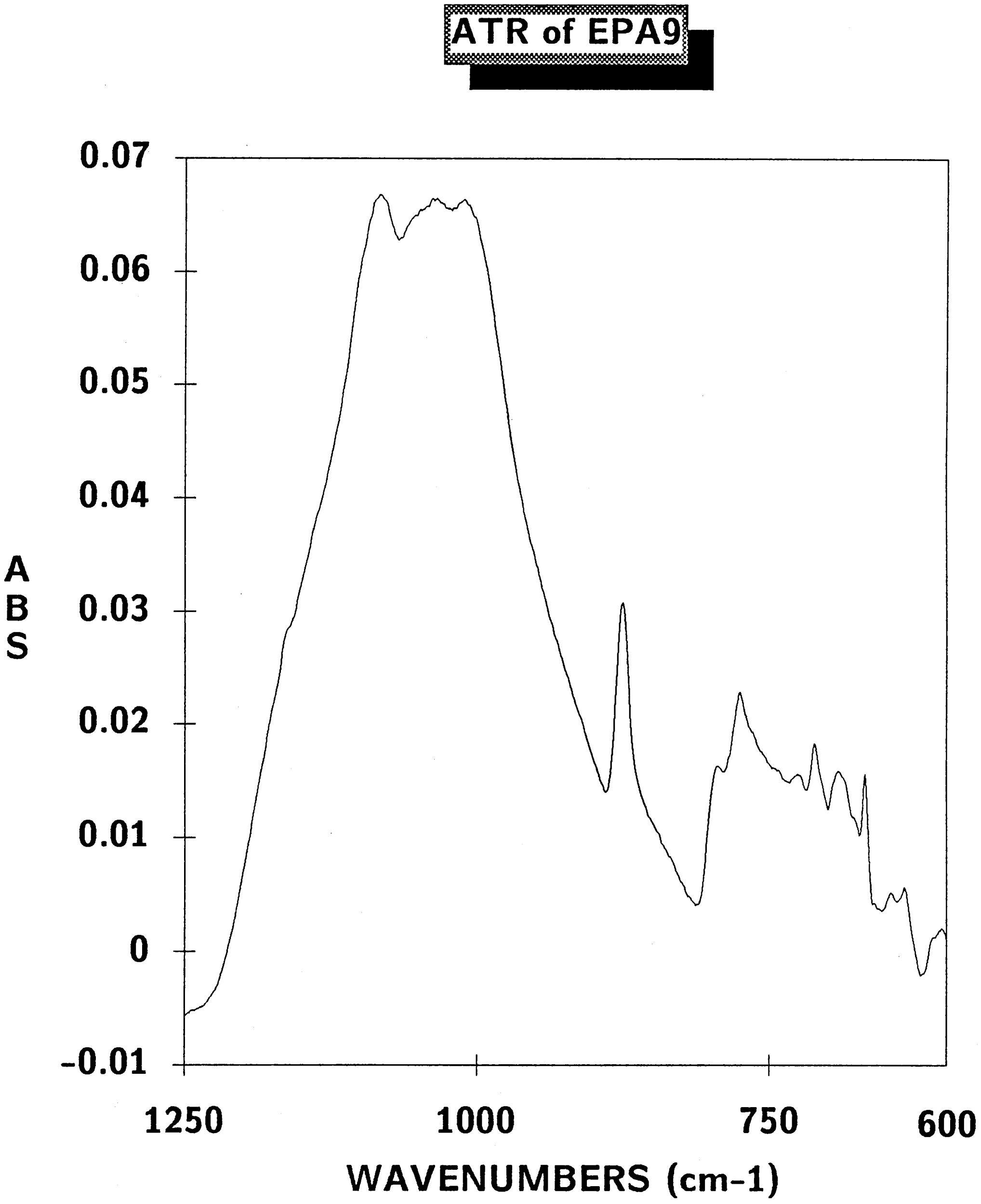

ATR of EPA4
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
-0.02
-0.04
-0.06
ABS
1250
1000
750
600
WAVENUMBERS (cm-1)


ATR of EPA5
0.2
0.1
0
-0.1
ABS
1250
1000
750
600
WAVENUMBERS (cm-1)


ATR of EPA6
0.16
0.12
0.08
0.04
0
-0.04
A
B
S
1250
1000
750
600
WAVENUMBERS (cm-1)


ATR of EPA7
0.2
0.15
0.1
0.05
0
-0.05
A
B
S
1250
1000
750
600
WAVENUMBERS (cm-1)


ATR of EPA8
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
-0.02
-0.04
ABS
1250
1000
750
600
WAVENUMBERS (cm-1)


ATR of EPA9
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
-0.01
ABS
1250
1000
750
600
WAVENUMBERS (cm-1)

VITA

KAILASH SWARNA

Candidate for the Degree of

Master of Science

Thesis: FTIR-ATR CHARACTERIZATION OF AQUIFER MINERALS

Major Field: Chemistry

Biographical:

Personal Data: Born in Madras, India, September 1, 1963, one of two children of S.V.M. Rao and Prema Rao.

Education: Graduated from DTEA High School, New Delhi, India, June 1980; received Bachelor of Science Degree from University of Madras, India, with a major in Chemistry in July 1983; completed requirements for the Master of Science Degree at Oklahoma State University in July, 1988.

Professional Experience: Graduate Teaching Assistant, Oklahoma State University 1985-1988; Graduate Research Assistant Summer 1986, Summer 1987; Member of American Chemical Society, and Phi Lambda Upsilon, Honorary Chemical Society.