HARTLEY TRANSFORM BASED ALGORITHM FOR THE QUALITATIVE AND QUANTITATIVE ANALYSIS OF MULTI-COMPONENT MIXTURES WITH THE USE OF EMISSION EXCITATION MATRICES.

by

George Asimopoulos

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Chemistry

APPROVED:

R. E. Dessy, Chairman

J.G. Mason

G.L. Long

J.S. Merola

J.M. Tanko

October 1992

Blacksburg, Virginia

HARTLEY TRANSFORM BASED ALGORITHM FOR THE QUALITATIVE AND QUANTITATIVE ANALYSIS OF MULTI-COMPONENT MIXTURES WITH THE USE OF EMISSION EXCITATION MATRICES.

by

George Asimopoulos

(ABSTRACT)

Rapid advances in computer technology over the last few years and their integration into analytical instruments have led to tremendous increases in data collection rates. The need for tools to assist analytical chemists, and especially spectroscopists, in their task of interpreting such vast quantities of data is immediate.

This work focuses on the development of an algorithm based on an alternative to the Fourier transform, the Hartley transform, for the qualitative and quantitative analysis of multi-component mixtures using Excitation Emission Matrices. The algorithm involves the reverse search of a compressed reference spectral library for the identification of possible components of the mixture and the method of Non-Negative Least Squares for the quantification of the components.

A number of techniques for pre-processing of three dimensional fluorescence spectra along with several spectral encoding methods for the compression of the spectra were investigated. Both simulated and real data collected with a

fluorescence spectrophotometer were used in this study.

The algorithm proved capable of analyzing mixtures of five components with relative concentrations ratio of about 100:1 and significant spectral overlap. At the same time a compression ratio of about 10:1 for the spectra in the reference library was achieved.

Finally, a library of three dimensional fluorescence spectra of some aromatic and poly-aromatic hydrocarbons was developed to be used with the algorithm. Such a library, along with the algorithm, provides a tool for the quick and simple qualitative and quantitative determination of mixtures of aromatic and poly-aromatic hydrocarbons.

ACKNOWLEDGEMENTS

I would like to thank Dr. R.E. Dessy and his wife Lee Dessy for all the help and attention they offered me during those years in graduate school. Also I would like to thank everyone in the Laboratory Automation and Instrument Desing group for their help and the unique envoroment they create.

Also I would like to thank Dr. Ehrich and Dr. Watson for their useful discussions, the Chemistry Department electronic shop for their help on maintening the spectrophotometer and the Perkin-Elmer Corporation for donating the equipment and their financial support.

I would like to offer this work to my parents in order to express appriciation for all the love and support they offered me through out my life. [Θα ηθελα να προσφερω αυτη την εργασια στους γονεις μου για να τους ευχαριστισω για την αγαπη και την συμπαρασταση που μου προσφεραν.] Finally, I would like to thank my brother Nikos for all his love and support, which made my stay in school so much easier. Thank you.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

I. INTRODUCTION

Rapid advances in computer technology over the last few years and their integration into analytical instruments have led to tremendous increases in data collection rates. The need for tools to assist analytical chemists and especially spectroscopists in their task of interpreting such vast quantities of data is immediate.

The majority of samples submitted for chemical analysis are mixtures. It is possible to separate the components of a mixture physically before the analysis, by filtration, extraction, or chromatography. However, these techniques are not always successful and they add considerable effort to the analysis.

A number of analytical spectroscopic techniques used to examine pure compounds produce signals suitable for mixture analysis, but the output is often difficult to interpret. In some cases a combination of such simple analytical techniques with sophisticated mathematical and statistical data processing methods, which fall into the area of Chemometrics, can provide an easy, fast and very informative way to analyze mixtures.

With the amazing processing speeds that computers have achieved, the selection of methods available to Chemometricians is enormous. Algorithms and procedures, that only few years ago where unrealistic to use because of

the long manual computational times required, now only take a few minutes or less to execute on computers found in every analytical laboratory. Methods for spectral enhancement, deconvolution, and data reduction such as: Maximum Likelihood, Linear and Multiple Regression, Curve Fitting, Pattern Recognition, Factor Analysis, and Maximum Entropy approaches, Expert Systems, Monte Carlo Calculations, Library Searching and Modeling are available.

Fluorescence, combining high sensitivity and multiple dimensions for selective information, is a technique well suited for mixture investigation. A combinations of fluorescence spectroscopy with a well implemented Chemometrics algorithm can be a very successful approach to the analysis of multi-component mixtures.

This work presents an algorithm developed for the analysis of mixtures of fluorescence compounds, which utilizes a compressed reference spectral library of three dimensional fluorescence spectra. The algorithm achieves complete qualitative and quantitative analysis of an unknown mixture.

II. HISTORICAL

A number of different approaches have been investigated over the last few years in an attempt to provide spectroscopists with tools for the analysis and interpretation of the vast quantities of data which modern instruments are capable of generating. Those methods vary from simple linear regression methods to those which involve computations and computer analysis that challenge today's computer hardware and software technologies. No one approach has been proven suitable and successful for all situations.

The next section takes a closer look at some of the approaches reported over the years. First, methods developed to assist the interpretation of fluorescence spectra, especially two-dimensional spectra are examined. Then methods designed to analyze multicomponent systems are explored. Finally, spectral library search, and data compression algorithms are reviewed.

A. FLUORESCENCE SPECTROSCOPY

Fluorescence spectroscopy is a widely used analytical technique in chemistry and related fields.[1] The measurement of fluorescence is inherently a multi-parameter technique because even the simplest measurement involves variation of more than one parameter.

For example, the measurement of the luminescence (fluorescence or phosphorescence) from a sample involves the simultaneous use of an excitation wavelength, λ_{ex}, and the corresponding emission wavelength, λ_{em}. The measured luminescence intensity, I_L, can then be represented as a function of λ_{ex} and λ_{em}

$$I_L = f(\lambda_{ex}, \lambda_{em}) \quad (1)$$

Selective measurements of individual luminophores in simple or even complex mixtures can be achieved by varying these two parameters.

Data collected by measuring the fluorescence intensity for a range of excitation and emission wavelengths is usually represented in a form of a numerical matrix, commonly referred to as an Excitation-Emission Matrix or EEM, with typical dimensions of 64x64 data points. Figure 1 shows a typical Excitation-Emission Matrix.

Early methods for acquiring an EEM involved collecting separate emission spectra at several excitation wavelengths and combining the individual spectra to form the three-dimensional profile. This method is obviously time consuming.

In 1975 I.M. Warner et al.[2] reported the first video-fluorometer, VF, which dramatically reduces the amount of time required in order to acquire an EEM. This instrument

TYPICAL EXCITATION-EMISSION MATRIX
ANTHRANILIC ACID (7.3E-6 M)

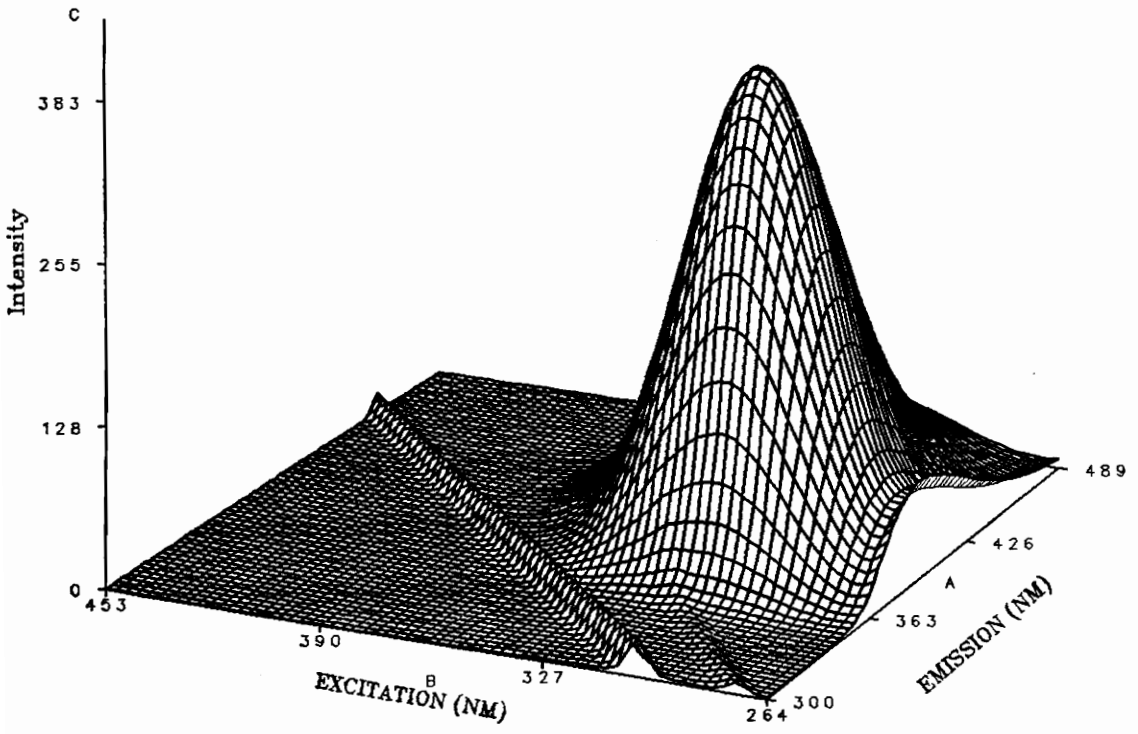


Figure 1. Typical Excitation Emission Matrix.

uses polychromatic excitation and a silicon intensified target vidicon detector (television camera) to acquire data in matrix format without mechanical scanning. The VF can acquire 64 emission spectra generated at 64 excitation wavelengths in about one second. J.B. Zung et al.[3] reported some of the changes which have occurred in the instrumentation for acquiring an EEM since that initial report of the video-fluorometer, but the significance of this first report can only be realized by examining the exponentially increased number of papers related to EEMs since that time. The purpose of the use of EEMs in those papers vary from the analysis of multi-component mixtures,[4] to bacteria identification,[5] and the investigation of the complexation properties of aromatic compounds.[6]

The video-fluorometer combined with computerized data-acquisition systems made data processing the time limiting step in the analysis of EEMs. Researchers have developed numerous computer aided spectral interpretation methods which also reduce the amount of data that needs to be stored. Pattern recognition is a spectral interpretation and data reduction method often employed.

T.M. Rossi and I.M. Warner developed a pattern recognition method which operates in the frequency domain of the Fourier transformation, using cross-correlation analysis[7]. In their approach, EEMs are represented as two

dimensional functions. The correlation of two functions f(x,y) and g(x,y) is defined as

$$f(x,y) \cdot g(x,y) = \int_{-\infty}^{\infty}\int f(\alpha,\beta)\, g(x+\alpha, y+\beta)\, d\alpha d\beta \qquad (2)$$

where α and β are generally referred to as shift parameters and the symbol "$\cdot$" denotes correlation. Since the discrete analog of Equation (2) is time consuming to implement on a computer, the calculation of the correlation functions is carried out in the frequency domain of the Fourier transform, FT, with the use of the correlation theorem of the Fourier transform. The theorem is defined by

$$f(x,y) \cdot g(x,y) \leftrightarrow F(u,v) \times G^{*}(u,v) \qquad (3)$$

where the symbol "$\leftrightarrow$" denotes a Fourier transform pair and the superscript "*" indicates the complex conjugate of the given function.

For the identification of an unknown spectrum the correlation functions of the unknown with a series of known spectra are calculated. Quantitative measurement of spectral differences within each unknown-known spectral pair are obtained by calculating three parameters : the sum, R, of the negative real coefficients of the correlation function, the sum, I, of the absolute value of the imaginary coefficients of the correlation function, and the intervector distance, IVD, between the Fourier transforms of

the two spectra. From the values of the three parameters, identification can be achieved by comparing these values with threshold values for correct spectral matches which previously have been estimated. Using this method the researchers were able to identify spectrally similar anthracene derivatives.

A method using the above pattern recognition algorithm has been developed by Chou-Pong Pau et al.[8] for bacterial fingerprinting. The method is based on the differences in enzyme content and activity of various bacteria. Bacterial cells mixed with carefully chosen fluorogenic enzyme substrates produce two-dimensional fluorescence spectrum, characteristic of the bacterium. In this approach, quantitative measurements of spectral differences are obtained by calculating the dissimilarity index, P,

$$P = (-P)(I)(IVD) \tag{4}$$

where R, I and IVD have the same definition as those given above. The researchers noticed that for a perfectly matched spectra pair these three parameters have a value of 0, which result in P = 0. Consequently, they found that the magnitude of P can serve as an index of spectral differences : the smaller the value, the closer the match. The researchers demonstrated the feasibility of their approach by accurately and quickly distinguishing six strains of

Escherichia coli. Classical methodologies require a time consuming series of complex tests before an identification of those bacteria could be achieved.

Further evaluation of the above mentioned pattern recognition algorithm by Chou-Pong Pau and I.M. Warner[9], using computer simulated data matrices and spectra acquired by a video-fluorometer, showed that background noise and other spectral characteristics, such as signal to noise ratio, peak width, etc., can significantly affect the results of the algorithm. These effects make the algorithm unattractive for a larger size spectral library search method.

The realization that noise can significantly affect the results of different algorithms led researchers to develop filtering techniques for EEMs. Although VF makes possible the acquisition of multiple scans of one spectrum in order to improve the signal to noise ratio, (S/N), it is known that the S/N ratio will improve proportional only to the square root of the number of scans,[2] therefore the improvement is not very significant.

M. Vicsek et al.[10] reported four time-domain filtering techniques for enhancing the information of two-dimensional fluorescence data. T.M. Rossi and I.M. Warner[11] reported filtering methods that operate in the frequency domain of

the Fourier transform for use with two-dimensional fluorescence data. Both groups failed to see any significant improvement in the quantitative value of the data after applying their methods. Filtering proved useful only when data was used for qualitative purposes.

An obvious approach for additional selectivity in luminescence measurements has been reported by I.M. Warner et al.[12] He described luminescence intensity I_L in terms of three parameters

$$I_L = f(\lambda_{ex}, \lambda_{em}, P) \qquad (5)$$

where P is one of a number of luminescence parameters including luminescence lifetime, light polarization, etc. Furthermore, more than three variables could be used to increase the selectivity of the measurement.

Matrix isolation and low-temperature fluorescence spectroscopy are two methods in which the sample is cooled to very low temperatures. In matrix isolation the sample is vaporized, mixed with an inert species which is a gas at room temperature (such as nitrogen or argon) and deposited on a cold surface. This technique has been successfully used for the characterization of mixtures of polycyclic aromatic hydrocarbons (PAHs)[13]. In low-temperature fluorometry the liquid sample is rapidly frozen to temperatures of 77 K or less. This technique has also

proved successful in the characterization of PAHs mixtures[14], but could not identify individual components of the mixture. Fluorescence spectra obtained under these conditions consist of bands which are much narrower than those observed under normal conditions. That is because the sample molecules are isolated and occupy strictly oriented positions in the low temperature matrix, thus their vibronic components become very sharp. This line-narrowing phenomenon is commonly called the "Shpol'skii effect"[15].

Until recently the analytical applications of fluorescence spectroscopy were limited to the use of the steady-state intensities, because of the complex and expensive instrumentation required for time-resolved measurements. For those measurements the sample is excited with a sinusoidally modulated light, resulting in fluorescence emission that is modulated at the same frequency, but phase-shifted and demodulated as function of the fluorescence lifetime, τ, of the fluorophores. Phase-resolved detection of the fluorescence signal produces a time-independent, phase-resolved fluorescence intensity, PRFI. D.W. Millican and L.B. McGown[16] described such a method which incorporates fluorescence lifetime selectivity into EEM data. The resulting data format is referred to as phase-resolved EEM or PREEM. In PREEM, fluorescence intensity is not only a function of the excitation and

emission wavelength but also a function of fluorescence lifetime. The equation given by Millican and McGown for the phase-resolved fluorescence intensity, PRFI is

$$PRFI = A' m_{ex} m \cos(\phi_D - \phi) \quad (6)$$

where A′ is the steady-state fluorescence intensity, m_{ex} is the modulation depth (ac/dc ratio) of the exciting light, m is the ratio of the emission modulation depth to m_{ex}, ϕ is the phase shift of the emission beam relative to the excitation beam, and ϕ_D is the phase of the detector, which can be set to any value between 0° and 360°. In equation (6), the demodulation m, and the phase shift ϕ, are related to fluorescence lifetime by

$$m = [(\omega\tau)^2 + 1]^{-1/2} \quad (7)$$

and

$$\phi = \arctan(\omega\tau) \quad (8)$$

where ω is the angular modulation frequency, or 2π times the linear frequency, f.

In two component mixtures, where components had unequal contributions to the total intensity, D.W. Millican and L.B. McGown[17] found that PREEMs were superior to steady-state EEMs in resolving the individual spectra of a two components with the use of multiway analysis. P.M Ritenour Hertz and L.B. McGown[18] reported the first application of phase-

resolved fluorescence spectroscopy for spectral fingerprinting. A set of petroleum-based lubricants (petrolatums) were characterized and discrimination between different petrolatum samples was achieved.

J.R. Lakowicz et al.[19] recently presented a method for the resolution of multi-component fluorescence emission using frequency-dependent phase angle and modulation spectra. The researchers used phase angle spectra and modulation spectra of the mixture, measured over a range of suitable light modulation frequencies and emission wavelengths. In this method the sample is assumed to consist of a mixture of fluorophores, each of which displays a single exponential decay time. The expression for the time-dependent emission of each wavelength (λ) is a multi-exponential decay

$$I(\lambda,t) = \sum \alpha_i(\lambda)\, e^{-t/\tau_i} \qquad (9)$$

where the pre-exponential factor ($\alpha_i(\lambda)$) depends on emission wavelength. The decay times (τ_i) are assumed to be characteristic of each component in the mixture and to be independent of wavelength for each component. The frequency domain data consist of phase ($\phi_{\omega\lambda}$) and modulation ($m_{\omega\lambda}$) values, each measured over a range of light modulation frequencies (ω) and emission wavelengths (λ). The collected

data consist of multiple sets of phase and modulation spectra, each measured at a single modulation frequency. The data is then analyzed by nonlinear least-squares analysis to recover the emission spectra of the individual fluorophores and the associated decay times.

Another luminescence-based selectivity parameter that has been used by F.V. Bright[20] to increase the selectivity of luminescence measurements, is what is called Fluorescence Anisotropy Selective Technique (FAST). This is based on rotational diffusion rate effects. The excitation of a random distribution of fluorescent molecules by linearly polarized light results in the preferential excitation of those molecules whose absorption dipoles are oriented along the polarization axis. Because of this selection, a nonuniform distribution (anisotropy) of excited-state molecules is generated. This induced anisotropy decays as a function of time because of molecular motion and solute-solvent interactions. The result of this decay of anisotropy is a randomization of the molecular emission dipole which manifests itself in a time-dependent depolarization of the resulting fluorescence. FAST is capable of recovering the individual spectral components in complex mixtures on the basis of differences in the rotational diffusion rates of the components.

From the above discussion it is obvious that much

effort has been put into increasing the selectivity of fluorescence measurements, thus making possible the use of the method for the analysis of complex mixtures. In the next section a number of algorithms developed for the interpretation of multi-component data will be presented.

B. MIXTURE ANALYSIS

The majority of samples submitted for chemical analysis are mixtures. It is possible to separate the components of a mixture physically before the analysis by filtration, extraction, or chromatography. However, these techniques are not always successful and they add considerable effort to the analysis. A number of instrumental techniques produce signals which can be used for the investigation of mixtures, both qualitatively and quantitatively. A number of methods have been developed to achieve exactly this goal. Some methods simply find the number of components in a mixture, others attempt to extract the spectra of the individual components, and yet some others attempt a quantitative analysis of the mixture. This section takes a closer look at some of those methods developed for the interpretation of multi-component data.

Most of the early reports attempt to provide the capability of component deconvolution, that is to extract the spectra of the individual components from the mixture

spectrum, or in some cases to simply find the number of components in the mixture.

As early as 1977, I.M. Warner et al.[21] reported a method for the qualitative analysis of multicomponent EEMs. The method analyzes the data matrix in terms of eigenvectors and eigenvalues. When the absorbance of the sample is less than 0.01,[22] the EEM data matrix, M, in the case of an r-component mixture can conveniently represented as

$$M=\sum_{i-1}^{r}\alpha_i x(i)y(i)^t \qquad (10)$$

where x(i) and $y(i)^t$ are the observed excitation and emission spectra of the i^{th} component and α_i is a concentration dependent parameter. For such a matrix M, there are several mathematical procedures available for the calculation of the eigenvectors, and the corresponding eigenvalues[23,24]. The number of eigenvectors associated with a large value eigenvalue, equals the number of independently emitting compounds in the mixture. Additionally, eigenvector analysis in the simple case of two component mixtures can extract the EEMs of the individual components. Another method using eigenvector analysis was recently reported by M. Kubista[25] where the individual components of the sample can be identified if two spectra of the mixture with different relative concentrations of the

components are available.

A similar method using target factor analysis, was reported by M. McCue and E.R. Malinowski[26] for the investigation of infrared spectra of multicomponent mixtures. For the case of an r-component mixture, the method requires a series of at least 2r mixtures containing the same components but in different concentrations, whose absorbance is measured at least in 2r wavenumbers. The method can give the number of absorbing species, test for the presence of suspected components in the series of mixtures. In the case where all r components have been identified from the target-testing procedure a quantitative analysis of each solution can be made.

The ratio method developed by T. Hirschfeld[27] for use with infrared data, and also reported by M.P. Fogarty et al.[28] for use with three dimensional fluorescence data, can determine the spectra of the individual components in related mixtures without prior knowledge of the constituents. The method requires r mixtures, where r is the number of components involved. Partly because each component must have a spectral region where it alone absorbs or emits, the method has been restricted to two and three component mixtures. M.P. Fogarty et al.[29] also developed a method which utilizes quenching as an aid in the ratio deconvolution of multicomponent fluorescence data. The

advantage of quenching is that apparent changes in the relative fluorescence intensity of the components, in order to create the series of mixtures required by the ratio method, can be accomplished without extensive sample preparation.

Researchers have also developed a number of methods for the enhancement of the information content of spectra of mixtures. F. Dousseau et al.[30] used the well-known method of spectral subtraction for the quantitative subtraction of water from the transmission infrared spectra of aqueous solutions of proteins.

Another very popular method for the enhancement of infrared spectra is the Fourier self-deconvolution method which was introduced by J.K. Kauppinen et al.[31] for resolving overlapped lines that can not be instrumentally resolved due to their intrinsic linewidth. The method uses one of the fundamental theorems of the Fourier transform, FT. The FT of the product of two functions is the convolution of the FTs of each. A measured spectrum, $S(\lambda)$, can be expressed as the convolution of a higher-resolution spectrum, $S'(\lambda)$, with a broadening function $G(\lambda)$,

$$S(\lambda) = S'(\lambda) * G(\lambda) \qquad (11)$$

where the symbol "*" denotes convolution. The $G(\lambda)$ function represents the broadening of the spectrum due to

instrumental effects and the fact that spectra are measured at finite resolution. Thus the problem of enhancing the resolution of a spectrum is reduced to the selection of an appropriate $G(\lambda)$ function, and then use the FT of that function to multiply the FT of the spectrum. Several functions have been suggested in the literature[32,33] for use in the Fourier self-deconvolution method, which result to line width reductions by factors of 3 or more. Disadvantages of the method is the high signal to noise ratio, S/N, required (S/N > 1000), and the side-lobes that appear along the sides of the peaks after the deconvolution.

The methods described so far do not assume prior knowledge of the composition of the mixtures under investigation. In cases where the identity of all the components or partial knowledge of the composition of the mixture is available, several methods have been reported which achieve quantitative analysis of the mixture. All of those methods assume a linear relationship between concentration and measured signal, absorption or emission.

In the simple case where two components are to be determined, measurements at two frequencies are needed to estimate the individual concentrations. For calibration of this type of system, two independent reference samples (samples with known composition) are necessary. This

approach can be extended to more components in the obvious way. An alternative, more robust approach would be to "over-determine" the system by using more than the minimum number of frequencies and reference samples, and use statistical and matrix procedures to estimate the solution. Most of these procedures offer a solution on the basis of minimizing the sum, R, of the squared residuals between the measured values, y_i, and those predicted by a theoretical model values, p_i,

$$R=\sum_{i-1}^{N}(y_i-p_i)^2 \qquad (12)$$

where N is the number of points in the two data sets. The predicted values, p_i, are the result of a theoretical model that needs to be developed and which includes the values from the reference samples. From the above expression, Equation (12), comes the commonly used name for this type of analysis, least-squares analysis.

The classical statistical procedure of multivariate least-square analysis is very often used for quantitative analysis of known-component mixtures. A comparison of several statistical and matrix methods for spectral quantitative analysis, such as multivariate least-squares, principal components, and partial least-squares, has been offered by M.P. Fuller et al.[34]. In the same article the

researchers also report their implementation of a partial least-squares method for use with infrared spectra. A successful application of this method was the quantitative analysis of samples of commercially available detergents[35].

Y. Li-shi and S.P. Levine[36] reported a least-square method applied to Fourier transform infrared (FT-IR) spectra for the quantitative analysis of multicomponent mixtures of airborne vapors of industrial hygiene concern. The least-square fit was successful in the quantitative analysis of mixtures of ambient air, with up to six component mixtures. They involved analytes found in hazardous waste sites, and could even handle those cases where there was strong overlap of the infrared spectral features.

A similar method for use with Near-IR Fourier transform (near-IR FT) Raman spectra has been reported by M.B. Seasholtz et al.[37], for the quantitative analysis of mixtures of unleaded gasoline, super-unleaded gasoline, and diesel. The difference in this approach was that instead of using pure compounds as their reference samples, the researchers used a calibration set of 29 mixtures composed of varying mass percentages of the three liquid fuels. The researchers also noticed something very important about the use of least-squares methods; by selecting different portions of the spectra to do the calculations there is a significant effect on the error of the suggested solution.

The same observation has been reported by I.M. Warner et al.[38] when the least-squares method was used for quantitative analysis of mixtures with EEMs. After further investigation, it was found that it is essential that only regions of the matrices with good signal to noise ratios be used. To achieve this, rather than using the entire 900 data points in each matrix, the researchers first reduced the number of data points to 36 by summing 25 neighboring points, and from those 36 points selected only those with the highest signal to noise ratios for the least-squares calculations. On the average, 10 points were selected and the method achieved the quantitative analysis of three component mixtures when all components were known.

When the complete qualitative composition of the sample is not known Ho et al[39]. developed a method called rank annihilation factor analysis, RAFA, which successfully has been used to predict the concentration of an analyte in an unknown sample in the presence of one or more chemical species unaccounted for in the calibration samples. This early rank annihilation method allowed quantitation in the presence of interferents, but it required the pure component response matrix for calibration and could only quantitate for a single analyte at a time (although multiple analytes were possible simply by repeating the mathematics for each analyte).

The method of rank annihilation is based on minimizing an appropriate function which involves the eigenvalues computed for the unknown and reference matrices. An extended method of rank annihilation, which allowed simultaneous multicomponent analysis has been reported by Ho et al.[40] The method was applied to a set of six component polynuclear aromatic hydrocarbon solutions by use of data acquired by a video-fluorometer in the form of an EEM. The calculated results for different compounds were greatly effected by the relative fluorescence of the analyte in the sample and the amount of spectral overlap.

A conceptual extension of the rank annihilation method of Ho et al. developed by E. Sanchez and B.R. Kowalski[41], is called the generalized rank annihilation method, GRAM. With the use of GRAM, it is possible to quantitate for multiple analytes in the presence of spectral interferents by using a single calibration sample. For example, if the LC/UV response matrices were measured under exactly the same conditions for a mixture containing eight components at known concentrations and for an unknown which contained some or all of the eight components, then it would be possible to use GRAM to obtain the concentrations of the mixture of these eight analytes as well as their isolated UV spectra and elution profiles. Besides carrying the problems of relative fluorescence intensity and spectral overlap of

RAFA, the complexity and time required to perform the calculations of GRAM make the method unrealistic as the search method for a large size spectral library.

The rank annihilation methods described above are applicable only to second-order bilinear data. The term second-order bilinear data describes data collected from instruments with the following two characteristics, (a) each sample yields a matrix of data, termed the response matrix, and (b) the rank of the response matrix for a pure chemical component is unity in the absence of noise. Examples of such techniques are liquid chromatography/ultraviolet (LC/UV), gas chromatography/mass spectrometry (GC/MS), and fluorescence excitation-emission matrices (EEM). Unfortunately, two very powerful instrumental methods, two-dimensional nuclear magnetic resonance (2D NMR) and two-dimensional mass spectroscopy (MS/MS), do not comply with the above requirements. B.E. Wilson et al.[42] developed a method called nonbilinear rank annihilation (NBRA), which they applied to 2D J-coupled NMR spectra. NBRA requires the pure component spectra for calibration, so that the direct multicomponent analysis and qualitative analysis advantages of GRAM are lost. B.E. Wilson and B.R. Kowalski[43] compared nonbilinear rank annihilation with three curve resolution methods for their abilities to accurately predict the concentration of an analyte in the presence of one or more

spectral interferents. Although, NBRA performed better than the three curve resolution methods, the results were inferior to multiple linear regression which was used as a referee method.

When the response of the pure components is available, an alternative to least-squares methods for multicomponent analysis are methods based on Kalman filtering. The Kalman filter is a recursive, digital filtering algorithm developed by R.E. Kalman[44] in the 1960s for engineering applications. It is a mathematical method which allows the estimation of system parameters, such as, the concentrations of fluorophores in an unknown sample, in the case of noisy and/or overlapped spectral responses. The Kalman filter algorithm is based on the generally valid assumption that the number of measurements is larger than the number of unknown concentrations. The following recursive structure is involved

$$NEWESTIMATE=OLDESTIMATE+CORRECTION \quad (13)$$

where "old estimate" is the estimate based on m measurements, the "new estimate" is the estimate based upon (m+1) measurements, while the "correction" is calculated on the basis of the new information supplied by the additional measurement.

H.N.J. Poulisse[45] describes a very attractive method,

because of the small number of computations required, based on the Kalman filter algorithm for multicomponent analysis of UV spectra. The author gives an example where the algorithm was able to accurately estimate the concentrations of the four components in a mixture where there was significant spectral overlap. C.B.M. Didden and H.N.J. Poulisse[46] further exploring the above method, show that in the situation where there are a number of candidate components, Kalman filters can be used to simultaneously determine the number of components present in the sample and their concentrations. If one of the components is not present in the sample, e.g. the i^{th} component, the filter will produce a very small value for the i^{th} coefficient, with respect to the coefficients for the other components. Although the Kalman filter algorithm is ideally suited for measurements corrupted by white noise, it is limited by the very small range of concentrations it can operate in. When the concentrations of the components vary more than one order of magnitude, the confidence intervals become so wide that the detection of minor components is impossible.

Kalman filter methods have also been used in two dimensional fluorescence spectroscopy. T.L. Cecil and S.C. Rutan[47] reported an algorithm based on the Kalman filter for the correction of spectral response shifts in overlapped

fluorescence spectra of polycyclic aromatic hydrocarbons. The algorithm corrects for variations in peak positions, peak intensity ratios, and fluorescence sensitivities, caused by changes in the solvent polarity, therefore giving significantly improved estimations for the concentrations of the components.

Besides the efforts to develop techniques for the analysis of mixtures, much emphasis in recent years has been placed on developing computerized methods for spectral interpretation. A number of different approaches has been studied over the years. Those vary from rule-based expert systems for the interpretation of infrared spectra,[48,49] to automated structure elucidation systems for the interpretation of two-dimensional NMR spectra,[50] to multi-parameter chi-square fitting procedures for ultraviolet spectra.[51] By far though, the most common computerized spectrum interpretation method employed today is library searching.

C. SPECTRAL LIBRARY SEARCH - DATA COMPRESSION

In this section, some of the most successful approaches for spectral interpretation with the use of spectral libraries, along with the search and the data reduction algorithms used to develop those libraries, will be

presented.

One very popular and widely used retrieval algorithm is the Probability Based Matching, PBM, algorithm developed by F.W. McLafferty et al.[52] for the identification of unknown mass spectra. The PBM is a statistical technique, which compares an unknown spectrum with a library reference compound and calculates the probability or 'Confidence Index', K, that the reference compound is present in the unknown. The calculations are based on the probability of individual peaks appearing in a spectrum. The library reference compound with the highest calculated 'Confidence Index' value represents the correct answer.

It is important to note that PBM uses a 'Reverse Search' technique. In reverse search, the system examines for the presence of the peaks of the reference spectrum in the unknown spectrum. In the opposite case or 'Forward Search', the system examines for the presence of the peaks of the unknown spectrum in the reference. The advantage of a reverse search system is that it can be used not only for the identification of pure compounds, but also for mixtures.

B.L. Atwater et al.[53] realizing that the confidence index, K, of the PBM algorithm gives only a qualitative indication of the probability that the retrieved compound represents a correct answer, developed an improved PBM system where the reference compounds are ranked according to

the predicted match reliability. This ranking substantially improves the performance of PBM, and the reliability value is especially helpful in avoiding the assumption that the best matching spectrum represents the correct compound when its spectrum is not actually in the reference library.

A 'Spectrum-stripping' technique has also been used to improve the identification of minor mixture components in matching unknown mass spectra using the PBM system.[54] In spectrum-stripping techniques, after one compound has been identified, its spectrum is subtracted from the unknown and the search continues. The process stops when the residual unknown spectrum contains only instrumental noise.

An automated mass spectrometry/mass spectrometry (MS/MS) search program has been developed by K.P. Cross and C.G. Enke,[55] which matches an unknown MS/MS spectrum against either primary or secondary spectra in a reference data base. The strategy of the program is first to eliminate the majority of candidate MS/MS spectra by prefiltering, and then using an intensity-based matching algorithm that retrieves an identical or structurally closely related reference compound (most of the time). The intensity-based algorithm was developed to recognize different kinds and degrees of similarity between the spectra, and for that reason uses seven match factors.

In the prefiltering step, the most significant peaks in

the unknown spectrum are ranked according to their increasing frequency in the data base. For every peak of the unknown spectrum, a subset of all the reference spectra that contain that peak, is created. The subsets of the two peaks with the lowest frequencies are ANDed together, resulting in a subset of the reference spectra which contains both peaks. An example of a logical AND operation can be seen in Figure 2. The process continues until all the subsets are used. The use of the least frequent peaks first, results in the majority of reference spectra being excluded in the first few AND operations, thus increasing the speed of the algorithm.

The methodology of selecting a subset of the spectral library to contain the spectra most similar to the unknown has also been used for infrared spectral libraries searching. J.M. Bjerga and G.W. Small[56] developed a method to decrease the time required to perform a standard library search based on principal components analysis. Principal components analysis calculates a new set of axes and coordinates which reduce the dimensionality of the original data space. The spectra are projected onto a principal plane where they are represented as a single point in a two-dimensional space. The angle of the point in the plane representing the unknown spectrum is determined, and only those library spectra with similar angles in the plane are

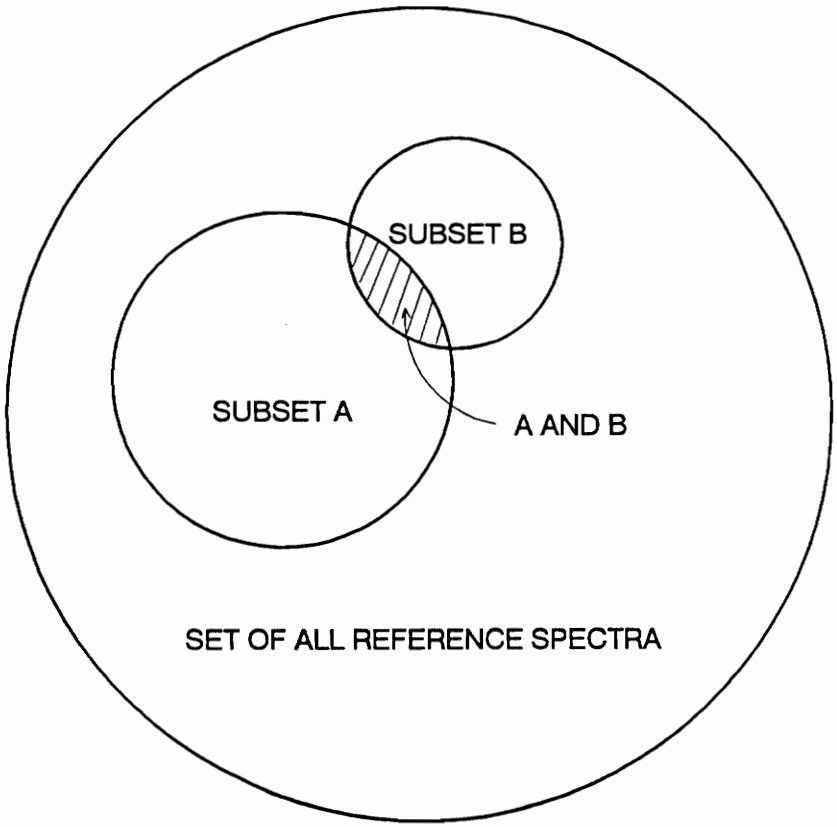


Figure 2. Logical AND operation.

selected and further searched with the use of the Euclidean distance or least-squares metric, Equation (12).

Library search techniques for structure identification have also been developed by O. Yamamoto et al.[57] for Nuclear Magnetic Resonance, NMR, spectra. For this purpose search files containing information taken from the full spectral patterns are created. In the search files peak information, including positions and intensities, as well as other search items such as molecular formula, molecular weight, etc. are stored. The method uses the ^{1}H-NMR area intensity rather than the peak height intensity for increased accuracy. The search is done simply by comparing the information in the search file for the unknown against the information in the search files of the reference spectra.

A similar spectrum compression algorithm that reduces the storage space required for infrared vapor-phase spectra by 95% with minimal loss of structural information content, has been described by R.A. Divis and R.L. White.[58] Fourier self-deconvolution is used to resolve overlapping bands, and a curve-fitting process is used to calculate and store intensity, location, and width of identified absorbance bands. Spectra compressed and stored in the reference library with this algorithm have to be reconstructed from the compressed data prior to the library search process,

which is done simply by calculating the Euclidean distance between the unknown and reference spectra.

Although this compression algorithm achieves a high compression factor, it is not considered a good compression algorithm for the generation of spectral reference libraries because it does not allow the search to be carried on the compressed form of the spectra. Carrying the search on the compressed form of the spectra has the obvious advantage of requiring less time for the search to be completed.

Z. Zolnai et al.[59] described a data compression method for NMR data, where the important information is localized in a small fraction of the overall data block. The compression algorithm involves two steps : elimination of the background noise and logarithmic scaling of the data. For the background noise elimination step, a threshold value is calculated from the standard deviation of the noise, and each spectral point is compared with this value : points below the threshold value are zeroed, and points above are left intact. Sequences of zeros are then replaced by the leading zero itself and a number indicating the number of zeros in the sequence. In the logarithmic scaling step, the data points left intact in the previous step are replaced by their logarithmic value with a suitably chosen base.

The overall compression factor of this method depends on the distribution of zero sequences in the original data

file, and typically ranges from 5 to 100. The fact that the comparison of two spectra can not be carried on the compressed form of those data files, and the widely varying of compression factors achieved, make this compression method inappropriate for the construction of spectral libraries.

F. Ishihara[60] developed a very efficient and fast method for the compression of spectral data and search of spectral libraries. The method was demonstrated using three-dimensional fluorescence spectra of polycyclic aromatic hydrocarbons. The EEMs are transformed with the use of two-dimensional Hadamard Transform, the higher sequences of the transformation are discarded as they containing only noise, and the remaining lower sequences are clipped into a series of 1's and 0's using a zero-crossing clipping algorithm. In the library search process, the clipped pattern of the unknown is XORed (Figure 3) with the patterns of the reference spectra to yield the corect match.

Numerous other approaches have been studied for data compression and library search algorithms, but the great majority of those are designed for specific applications, like the QUEST system of J.I. Garrels[61] for two-dimensional gel electrophoresis. The QUEST system automatically detects, resolves, and quantifies the spots that make-up the protein patterns on the two-dimensional gels. Those spots

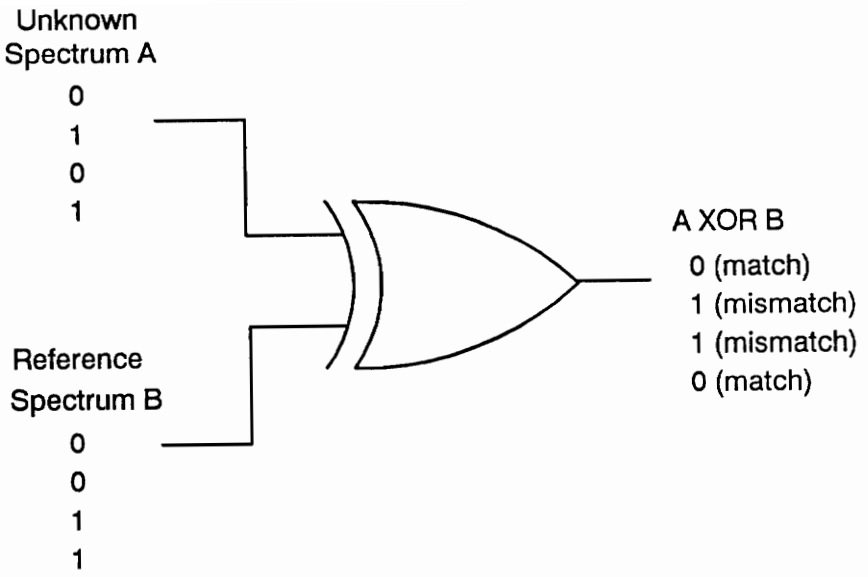


Figure 3. Logical XOR operation.

can then be entered into the QUEST database, which can be searched with four different matching algorithms for the analysis of an unknown gel. More details for the construction and analysis of protein databases using the QUEST system are given by J.I. Garrels and B.R. Franza, Jr. in another report.[62]

Data compression is also a very important issue in other areas, like image processing and speech processing, where large amounts of data must be compiled and analyzed. In most cases, methods from those fields can directly applied to spectral data compression. Such a case is described by I.E. Alguindigue and R.E. Uhrig[63] where neural networks are used to compress spectral signatures. Although the compression ratio achieved in this study was only 2 to 1, the results were very encouraging, pointing out the future feasibility of such an approach.

In closing this historical section it is important to emphasize that heretofore (1) the use of a spectral library of pure compounds for the analysis of pure compound unknowns has been successfully implemented, and (2) the use of a spectral library of mixtures of different compositions for the analysis of mixtures has also been successfully implemented, but the use of a pure compound spectral library for the the analysis of mixtures has never been reported. This is the goal of this research.

III. THEORETICAL BACKGROUND

A. FLUORESCENCE SPECTROSCOPY

In a conventional fluorescence measurement experiment, the sample is irradiated with monochromatic light which produces molecules in the excited state. As these molecules return to the ground state, light with a characteristic wavelength distribution is emitted, known as the fluorescence emission spectrum. A fluorescence excitation spectrum is the fluorescence intensity as a function of the absorption wavelength used to move the molecules to the excited state.

These light absorption and emission processes are very nicely illustrated by the energy-level diagram suggested by A. Jablonski in 1935.[1] In Figure 4, the ground, S_0, first, S_1, and second, S_2, electronic states along with the absorption and fluorescence processes, are shown in a simplified version of the original Jablonski diagram. During light absorption, molecules usually are excited to some higher vibrational level of either S_1 or S_2, followed by a rapid relaxation to the lower vibrational level of S_1. This relaxation process is called internal conversion and is represented by the broken lines in Figure 4.

The transitions between the various energy levels in

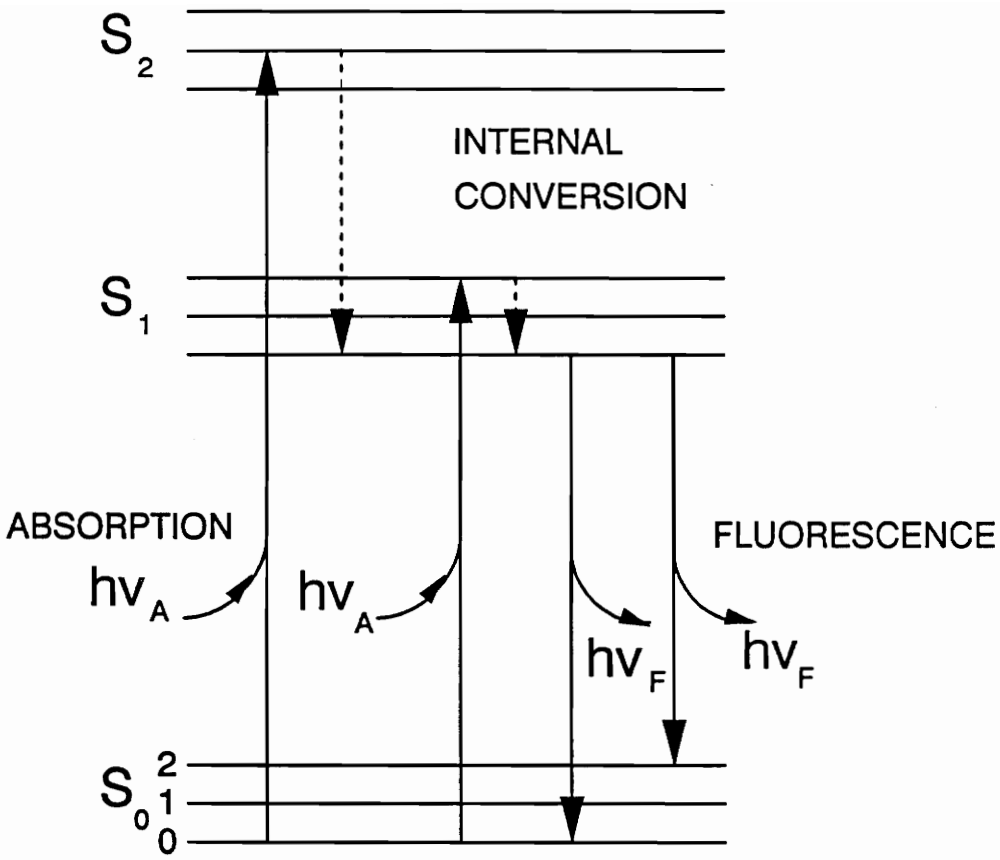


Figure 4. Simplified Jablonski diagram.

the Jablonski diagram are represented by vertical lines, a presentation chosen by Jablonski to illustrate the instantaneous nature of those transitions. The light absorption process occurs in about 10^{-15} sec, the internal conversion usually occurs in 10^{-12} sec, and fluorescence lifetimes are typically near 10^{-8} sec.

The simplicity of the diagram is explained by the Franck-Condon principle, which states that[1] " ... *the time required for an electronic transition is negligible compared with that of nuclear motion* ... " This is also the source of the mirror image rule of fluorescence[1] which states that the fluorescence emission spectrum appears as the mirror image of the absorption spectrum, specifically the absorption representing the S_0 to S_1 transition. Also, since the internal conversion is so fast, emission spectra are usually independent of the excitation wavelength.

Another important parameter of fluorescence spectroscopy which needs to be described is quantum yield. This parameter can best be illustrated by reference to the modified Jablonski diagram, Figure 5. In this diagram increased attention is directed to those processes responsible for the return to the ground state. In particular, two parameters are important, (1) the emissive rate of the fluorophore, Γ, which is the rate at which the

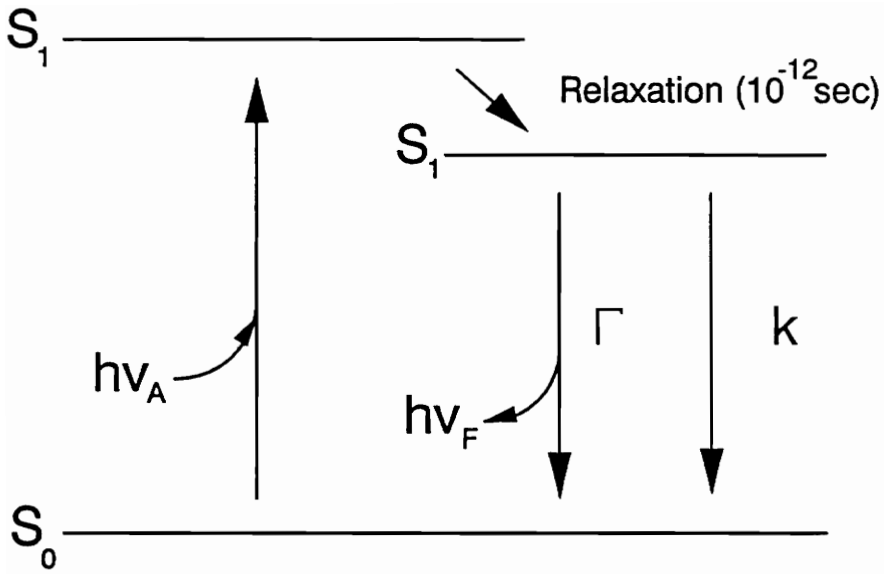


Figure 5. Modified Jablonski diagram.

excited molecules return to the ground state through emission of radiation, and (2) the rate of radiationless decay, k, which is the rate at which the molecules return to the ground state without radiation emitting processes, such as thermal and solvent relaxation.[47]

The fluorescence quantum yield, ϕ, is the ratio of the number of photons emitted to the number of photons absorbed by the fluorophore. Since the number of photons emitted is proportional to Γ and the number of photons absorbed is proportional to the sum $(\Gamma+k)$, the quantum yield is given by

$$\phi = \frac{\Gamma}{\Gamma + k} \qquad (14)$$

The quantum yield can be close to unity if the radiationless rate is much smaller than the rate of radiative decay (that is $k \ll \Gamma$). This is usually not the case, and typical values for the quantum yields of many compounds are much lower than unity. Table I shows the quantum yield values for some aromatic compounds.[64] It can be seen that these values vary widely and thus quantum yield is very a important parameter when determining the fluorescence intensity of a sample.

At low concentrations, when the absorbance of a sample is less than 0.01, the intensity of the fluorescence emitted at a given wavelength is directly proportional to the amount of light absorbed, and therefore it is also proportional to

Table I. Quantum yield values of some aromatic compounds.

Compound	Solvent	Quantum Yield
Anthracene	Benzene	0.26
	95% EtOH	0.27
Acridine	Ethanol	0.82
	95% EtOH	0.83
Fluorescein	H_2O-NaOH	0.93
9-Aminoacridine	Ethanol	0.99
	EtOH-HCl	1.00
	Water	0.98
Quinine sulphate	N H_2SO_4	0.54
9,10-Dichloro-anthracene	Benzene	0.65
1,8-Diphenyl-1,3,5,7-octa-tetraene	Benzene	0.15
Perylene	Benzene	0.89
1,4-Diphenyl-1,3-butadiene	Cyclohexane	0.44
Rhodamine B	Ethanol	0.97
	EtOH-HCl	1.00

the concentration of the analyte in the sample solution, following Beer's law. For a sample containing a single emitting species the fluorescence intensity, I, can be given to an adequate approximation by[22]

$$I = 2.303 I_0 \phi \epsilon b c \qquad (15)$$

where I_0 is the intensity of the incident radiation, ϕ is the quantum yield, ϵ is the molar extinction coefficient, b is the pathlength of the sample cell and c is the concentration of the fluorophore in the sample solution.

In an Excitation-Emission Matrix, M, each element, m_{ij}, which represents the fluorescence intensity at wavelength λ_i that was generated by excitation at wavelength λ_j, can be expressed by

$$m_{ij} = 2.303 \phi I_0(\lambda_i) \epsilon(\lambda_i) \gamma(\lambda_j) \delta(\lambda_j) b c \qquad (16)$$

where $\gamma(\lambda_i)$ reflects the dependence of I on the monitored emission wavelength and $\delta(\lambda_j)$ is a parameter which incorporates instrumental artifacts like sensitivity and signal collection geometry.

Combining the terms in Equation (16), based on the dependance of the variables on excitation or emission wavelengths, results in the simple expression

$$m_{ij} = \alpha x_i y_j \qquad (17)$$

where α is a scalar equal to $2.303\phi bc$, x_i is the excitation

term combining the excitation wavelength related variables, and y_j is the emission term, combining the emission wavelength related variables.

When the x_i and y_j are properly sequenced, the two arrays are representations of the excitation, **x**, and emission, **y**, vectors (spectra) of the fluorophore respectively. Since the excitation profile is independent of the monitored emission wavelength, and the excitation profile is independent of the monitored emission wavelength, the matrix M can be expressed as the vector product of the excitation and emission vectors, **x** and **y**, multiplied by the scalar concentration term α

$$M=\alpha xy^T \tag{18}$$

In a sample containing r fluorescent compounds, assuming again low concentrations for all components, the matrix M is the sum of the EEMs of the individual components. Thus, the r component matrix can be expressed as

$$M=\sum_{k-1}^{r}\alpha_k x_k y_k^T \tag{19}$$

Further simplifying Equation (19), one can combine the excitation and emission vectors, **x** and **y**, for each component into a matrix form, M_k, and write the mixture matrix, M, as

being the sum of the individual standard matrices of the components, each multiplied by a relative concentration factor, β_k

$$M=\sum_{k-1}^{r}\beta_k M_k \qquad (20)$$

where β_k is the concentration of the component k in the mixture divided by the concentration of the component k in the standard matrix, and the notation $\beta_k M_k$ means that each element of M_k is multiplied by the factor β_k.

The assumption of low concentrations made previously is very important if the above representation of the EEM of a mixture is to be pertinent. At higher concentrations of the analyte the relation between fluorescence intensity and concentration, Equation (15), becomes non-linear,[1] as can be seen in Figure 6. As the concentration of the analyte increases past point L_{upper}, measured fluorescence intensity drops because of reabsorption, i.e. part of the emitted light is absorbed back by the fluorophores. The lower limit, L_{lower}, shown in Figure 6, is the limit of detection for the particular spectrophotometer used for the measurements. Below this concentration, although the relationship between fluorescence intensity and concentration remains linear, accurate measurements of the fluorescence are not possible.

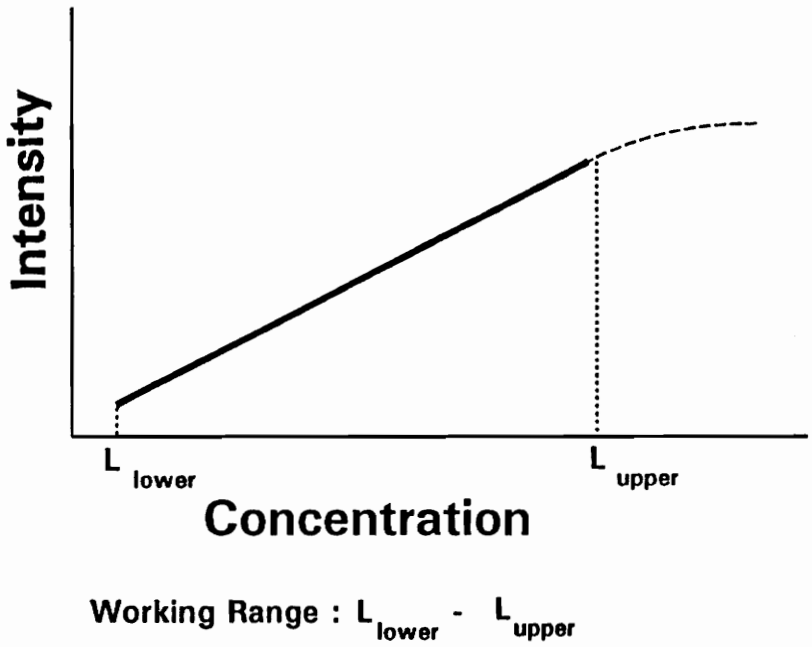


Figure 6. Relationship between fluorescence intensity and concentration.

The lower limit, L_{lower}, can be shifted towards lower concentrations of the analyte with the use of more sensitive instruments for the measurement of fluorescence, whereas the upper limit, L_{upper}, cannot be altered since reabsorption cannot be prevented. This poses no real problem for the algorithm which will be presented in the next section, since the problem most methods have is that they only work at higher concentration, and cannot handle low concentrations of analytes.

The range between L_{upper} and L_{lower} is the concentration range where the above equations apply, and thus it will be the concentration range which will be implied for the remaining of this work. This range is typically several orders of magnitude wide. The exact limits, though, vary for different compounds, because of the differences in the values of quantum yield.

The representation of the EEM of a mixture, as the sum of the EEMs of the individual components, Equation (20), will be used in the following developments. The assumption of a linear relationship between fluorescence intensity and concentration of analyte, as well as the presence or absence of synergistic effects, will be further explored in following sections.

B. HARTLEY TRANSFORM

Transform methods, especially the transform developed by Joseph Fourier which carries his name, have found an enormous range of applications in chemistry.[65] Different spectroscopic techniques use the Fourier transform to convert a complex and confusing time sequence, created by the physical processes involved in those spectroscopies, into an interpretable spectrum. Typical examples of this type of spectroscopy are Fourier transform infrared spectroscopy, Fourier transform NMR spectroscopy, and even Fourier transform mass spectroscopy.

Fourier transform techniques are also a powerful aid to signal processing. Those techniques not only allow the convenient transformation between two different representations of the data, but also simplify mathematical operations on the data. Typical applications include the calculation of the frequency-domain spectrum from a discrete time-domain data set, Fourier self-deconvolution of overlapped peaks, even signal filtering.

Although Fourier transform has become the preferred method for those applications, it is not the only transform technique that can be used to achieve these results. The Hartley transform offers a conceptually simpler alternative to the Fourier transform.

The reason for the wide spread of the Fourier

transform, over any other transformation, is the development of the discrete fast Fourier transform, FFT, by J.W. Cooley and J.W. Tukey,[66] in 1965, which tremendously increases the speed of the calculations. The corresponding discrete fast Hartley transform, FHT, was developed much later, in 1984, by R.N Bracewell[67].

The Hartley transform, was introduced by R.V.L. Hartley[68] in 1942. In contrast to Fourier transform, it maps a real function of time, X(t), into a real function of frequency, H(v). There is a strong connection between the two transforms, as the following statement indicates: the Hartley transform is the real part of the Fourier transform minus the imaginary part.

The Hartley transform, just like Fourier, transforms a function from one domain, (e.g. time), to its reciprocal, (1/time = frequency). At this point, it is important to realize that a transform pair is simply an alternative representation of the information about the system, and which representation is chosen is entirely a matter of convenience.

The equations for the definition of the Hartley transform for a continuous function extended to infinity in both directions, along with its inverse transform (used to map the frequency function back into the time domain) are:

$$H(v) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(t)\, cas(2\pi vt)\, dt \qquad (21)$$

$$X(t) = \int_{-\infty}^{\infty} H(v)\, cas(2\pi vt)\, dv \qquad (22)$$

where $cas(2\pi vt)=\cos(2\pi vt)+\sin(2\pi vt)$.

These equations are very similar to those of the Fourier transform and its inverse:

$$F(v) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(t)\, e^{-i2\pi vt} dt \qquad (23)$$

$$X(t) = \int_{-\infty}^{\infty} F(v)\, e^{j2\pi vt} dv \qquad (24)$$

where $e^{j2\pi vt}=\cos(2\pi vt)+j\sin(2\pi vt)$, and $e^{-j2\pi vt}=\cos(2\pi vt)-j\sin(2\pi vt)$, which are known as Euler's formulas. The principal difference between the two definitions is that the real function $cas(2\pi vt)$ in the Hartley transform replaces the complex exponential term $e^{\pm j2\pi vt}$ in the Fourier transform pair. Although this does not seem like an important difference, the Fourier transform of a function is obviously a complex function. Complex arithmetic requires more operations than real arithmetic (a complex multiplication or division requires four operations, and a complex addition of subtraction requires two operations). Furthermore, complex

data arrays require double the memory storage of real data arrays. Therefore, the Hartley transform will be distinctly faster and use less computer resources than the Fourier transform in applications where large amounts of data need to be processed. Also, since the Hartley transform uses fewer operations to process a signal, the transformed data would have fewer roundoff errors. Those errors are introduced by the limited precision by which computers carry-out calculations.

An additional advantage of Hartley transform over Fourier transform, is that the inverse Hartley transform can be obtained by applying the same algorithm to its own output, thus regenerating the input data. This means that the same computer code can be used to compute the transform and its inverse.

The definitions given above for the Hartley and Fourier transform deal only with continuous variables. In real experimental systems, the data in not continuous and it does not expand to infinity in both directions. In those cases, the discrete forms of the transform pair for a set of N data points is defined as

$$H(v) = \frac{1}{N}\sum_{t=0}^{N-1} F(t)\,cas(2\pi vt/N) \qquad (25)$$

and

$$X(t)=\sum_{v=0}^{N-1} H(v)\,cas(2\pi vt/N) \qquad (26)$$

From these equations it is apparent that for the computation of the Hartley transform of an N-element data set, N^2 arithmetic operations would have to be performed (N computations for every one of N points). For large data sets this number becomes extremely large, making the calculation of the transform difficult and very time consuming. To overcome this difficulty, the fast Hartley transform, FHT, algorithm, uses a permutation process to bisect the data until data pairs are reached. Calculating the Hartley transform from such data pairs is trivial:

$$(a,b)\leftrightarrow(a+b,a-b) \qquad (27)$$

The idea behind the permutation process is that it is faster to split the data into pairs, compute the transform of the pairs using the above equation, and recombine these pairs to make the entire transform, rather than to compute the transform for the complete data set using Equation (25). It takes approximately Nlog(N) computations, rather than N^2, for the FHT algorithm to compute the transform of an N-element data set. A computer implementation of the FHT algorithm using the permutation process can be found in the literature.[69]

An important point to note about the discrete form of the transform equations is that it is simply a transform of a series of numbers sampled at equal intervals. There is no requirement that the two representations of the system be the time and frequency domains, respectively. Any series of numbers could be transformed, regardless if they represent a spectrum, a system response, a data matrix, or even if they are random numbers or noise. In these cases though, the interpretation of the results would be different, and a simple physical interpretation of the transform domain, e.g. frequency, may not exist.

Another important feature of transforms, and of course the Hartley transform, is that mathematical operators undergo transformation as well as data. This means that an equivalent calculation can be carried out in the transform domain in a different way to the procedure which would have been used for the original data. Frequently, this can be used to simplify complex calculations.

For example, an extremely important process in signal processing is that of convolution and deconvolution.[70] The complex calculation of the convolution of two functions in one domain is equivalent to simple multiplication in the other domain. Given two functions, h(t) and g(t), and their corresponding transforms, H(v) and G(v), a Hartley transform pair can be defined such that

$$g(t) * h(t) \leftrightarrow G(v)H(v) \tag{28}$$

where the symbol "*" denote convolution and the symbol "↔" denotes a transform pair. Therefore, to calculate the convolution of two time functions it is only necessary to transform them individually, perform a point by point multiplication of their transforms and inverse transform back to the time domain.

R.N. Bracewell[70] explains the properties of the discrete Hartley transform. Some of those properties, along with some interesting relations of the discrete transform, can be found in Table II. Certain properties in Table II are very important for this work and will further be explained.

Multiplication by a scalar: A very simple property of the Hartley transform important for this work is the multiplication of a given function by a scalar. Multiplying a function, f(t), by a scalar, results in the multiplication of the Hartley transform of the function by the same scalar:

$$\alpha f(t) \leftrightarrow \alpha F(v) \tag{29}$$

where F(v) is the Hartley transform of the function, and α is any real number. It was explained earlier in the discussion on the representation of the Excitation-Emission Matrices, Equation (17), that each element of an EEM can be represented as a vector product matrix, multiplied by a

Table II. Properties of the Hartley transform.

Theorem	Function f(t)	DHT H(v)
Reversal	$f(-t)$	$H(-v)$
Scalar product	$\alpha f(t)$	$\alpha H(v)$
Addition	$f_1(t) + f_2(t)$	$H_1(v) + H_2(v)$
Convolution	$f_1(t) \circ f_2(t)$	$N/2\ [H_1(v)H_2(v) - H_1(-v)H_2(-v) + H_1(v)H_2(-v) + H_1(-v)H_2(v)]$
Product	$f_1(t)f_2(t)$	$N/2[H_1(v)\circ H_2(v) - H_1(-v)\circ H_2(-v) + H_1(v)\circ H_2(-v) + H_1(-v)\circ H_2(v)]$
Derivative	$f'(t)$	$2\pi vH(-v)$
2nd Derivative	$f''(t)$	$-4\pi^2v^2H(v)$
Sum of sequence	$\sum_{t=0}^{N-1} f(t) = NH(0)$	
First value	$f(0) = \sum_{v=0}^{N-1} H(v)$	

concentration factor, α. Since the Hartley transform can operate on any series of numbers, even a matrix, assuming that f(t)=**xy**, it can therefore be shown that each element of the transformed matrix will be proportional to the same concentration factor, α, i.e. each element in the transformed EEM is proportional to the concentration of the fluorescence species.

Addition: The addition of two functions can be carried out in any of the two domains of the Hartley transform, i.e. adding two functions in one domain is equivalent of adding their Hartley transforms:

$$g(t)+h(t)\leftrightarrow G(v)+H(v) \qquad (30)$$

where g(t), h(t), and G(v), H(v), are any two functions and their Hartley transforms respectively. In the case of EEMs, it was shown earlier, Equation (20), that the EEM of a mixture is the sum of the EEMs of the individual components. Thus, application of the addition property of the Hartley transform means that the transformed matrix of the mixture equals the sum of the transformed matrices of the individual components.

Combination of these two properties of the Hartley transform, multiplication by a scalar and addition, in the case of an r component mixture with concentrations β_k, k=1,2...r, results in the following relation:

$$M=\sum_{k-1}^{r}\beta_k M_k \leftrightarrow H(M)=\sum_{k-1}^{r}\beta_k H(M_k) \qquad (31)$$

where M and M_k are the EEM of the mixture and the EEMs of the individual components, respectively, and the symbols H(M) and $H(M_k)$ represent their Hartley transforms. The significance of the above relation will be discussed later in the section on least-square analysis.

A final question that has to be addressed about Hartley transform is "what form does the spectrum take after it is transformed?" and "how does the transform effect white noise?" That will be better demonstrated with the use of appropriate examples. In the following examples, all the spectra and the corresponding Hartley transforms consist of 256 points. A computer program (Appendix A) written in PASCAL was used to calculate the Hartley transforms.

Since fluorescence spectra usually involve broad, rather featureless peaks, fluorescence spectra are commonly simulated with the use of Gaussian peaks. Figure 7 shows such a peak. It is apparent that the signal is spread over almost all the points of the spectrum, and only points at the two far sides of the spectrum seem to contain no signal (Although in theory, Gaussian peaks extend to infinity at both sides).

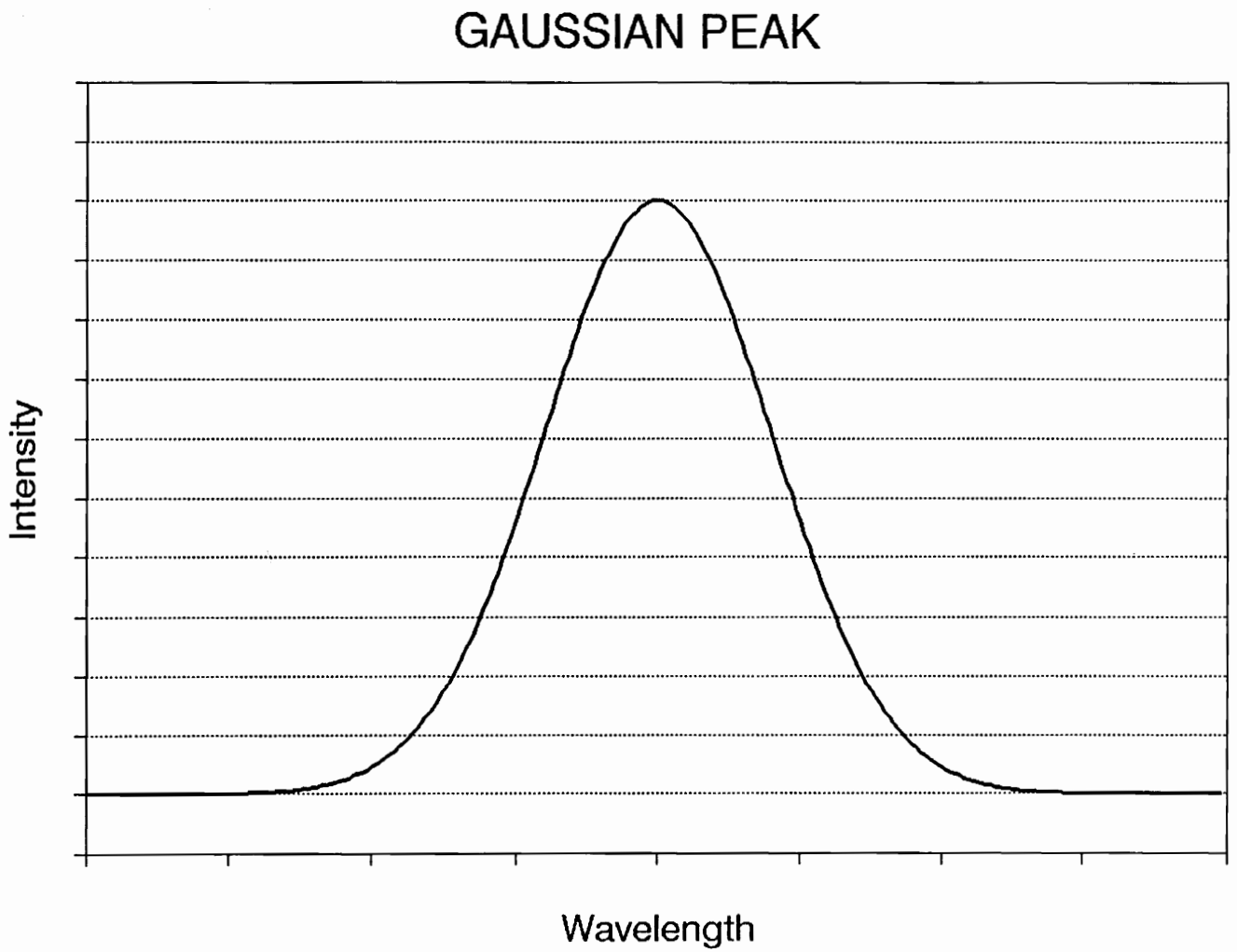


Figure 7. Gaussian peak.

The output of the fast Hartley transform of the above Gaussian peak is shown in Figure 8. In the representation used for the transform domain in Figure 8, points at the two sides represent low spatial frequencies, whereas points in the middle of the transform represent high spatial frequencies. It is obvious now that the signal has been redistributed, and in the transform domain only points representing low spatial frequencies contain significant amounts of signal.

The Hartley transform of an N-element data set representing broad, featureless peaks, shows most of the signal in the transform domain shifted towards the low spatial frequencies. The opposite is true with narrow peaks, showing full return to the baseline. The signal in the Hartley domain will be spread over most of the points, representing both low and high spatial frequencies.

This is not the case with white noise. Figure 9 shows a simulated spectrum containing only white noise. Since white noise contains all possible frequencies it would be expected that after the transformation points at low and high spatial frequencies would have equal amounts of signal. This is exactly what Figure 10 demonstrates. After the transformation, the noise remains spread over the entire spectrum, but at the same time it is compressed by a factor which equals the square root of the number of points in the

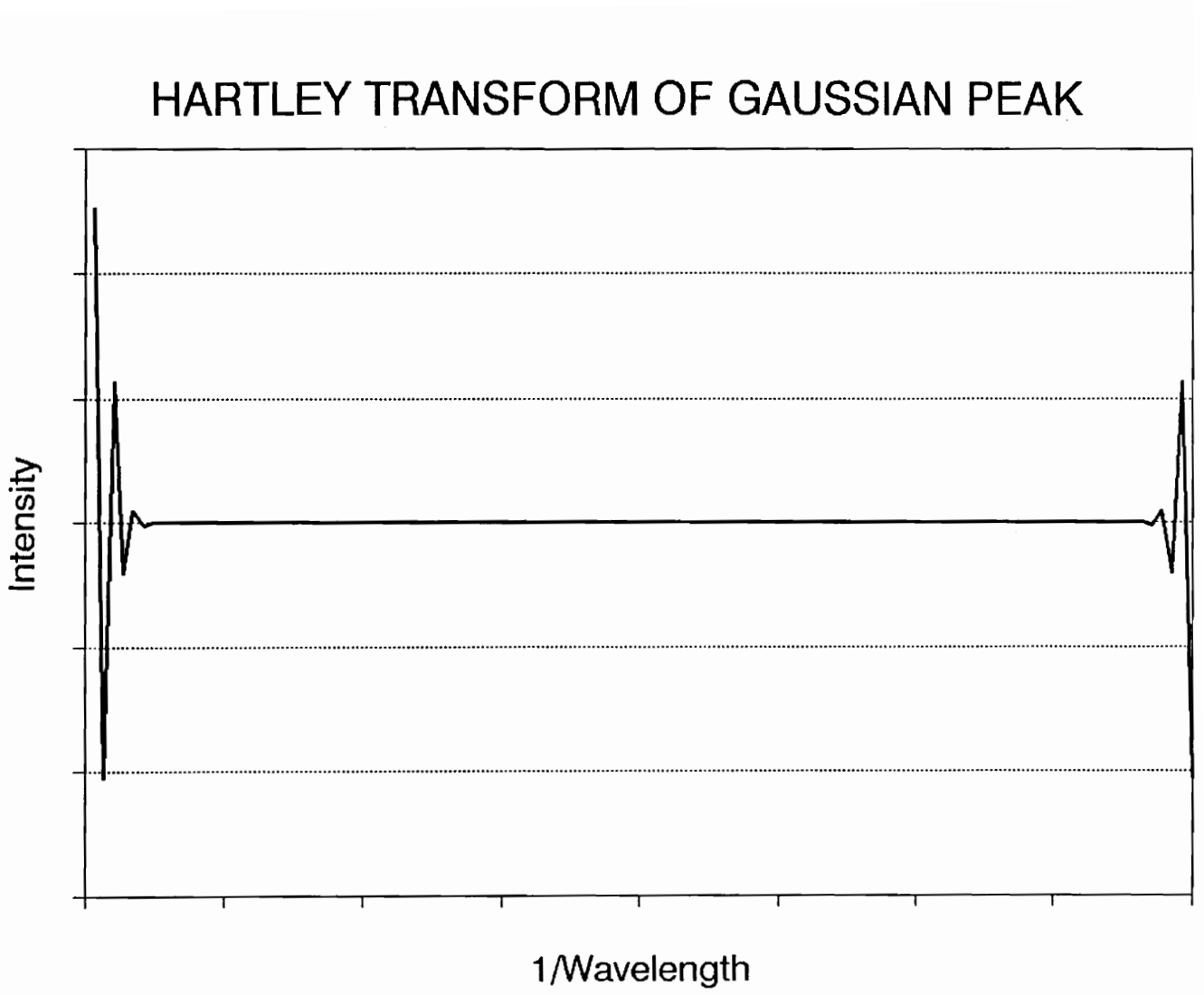


Figure 8. Hartley transform of a Gaussian peak.

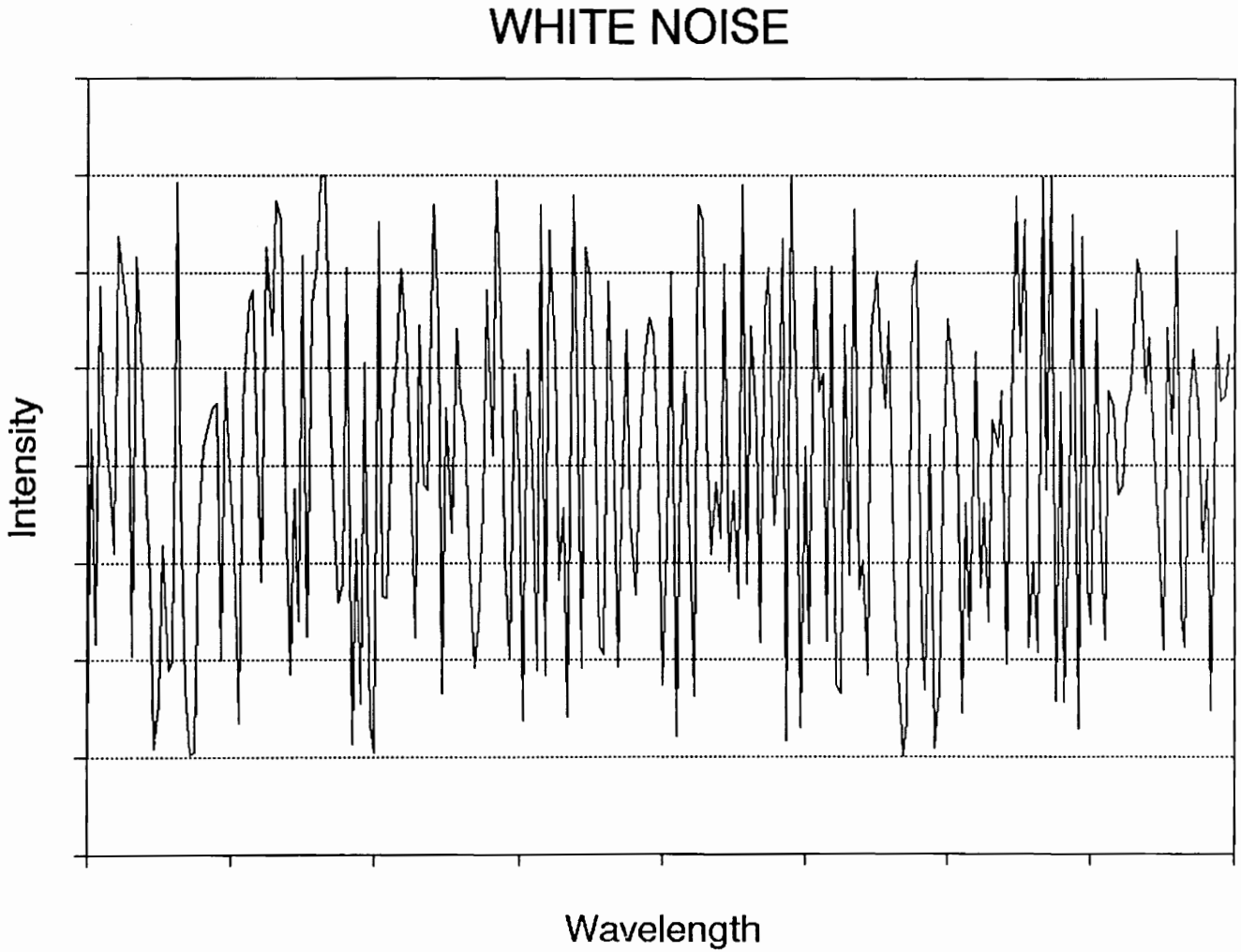


Figure 9. White noise.

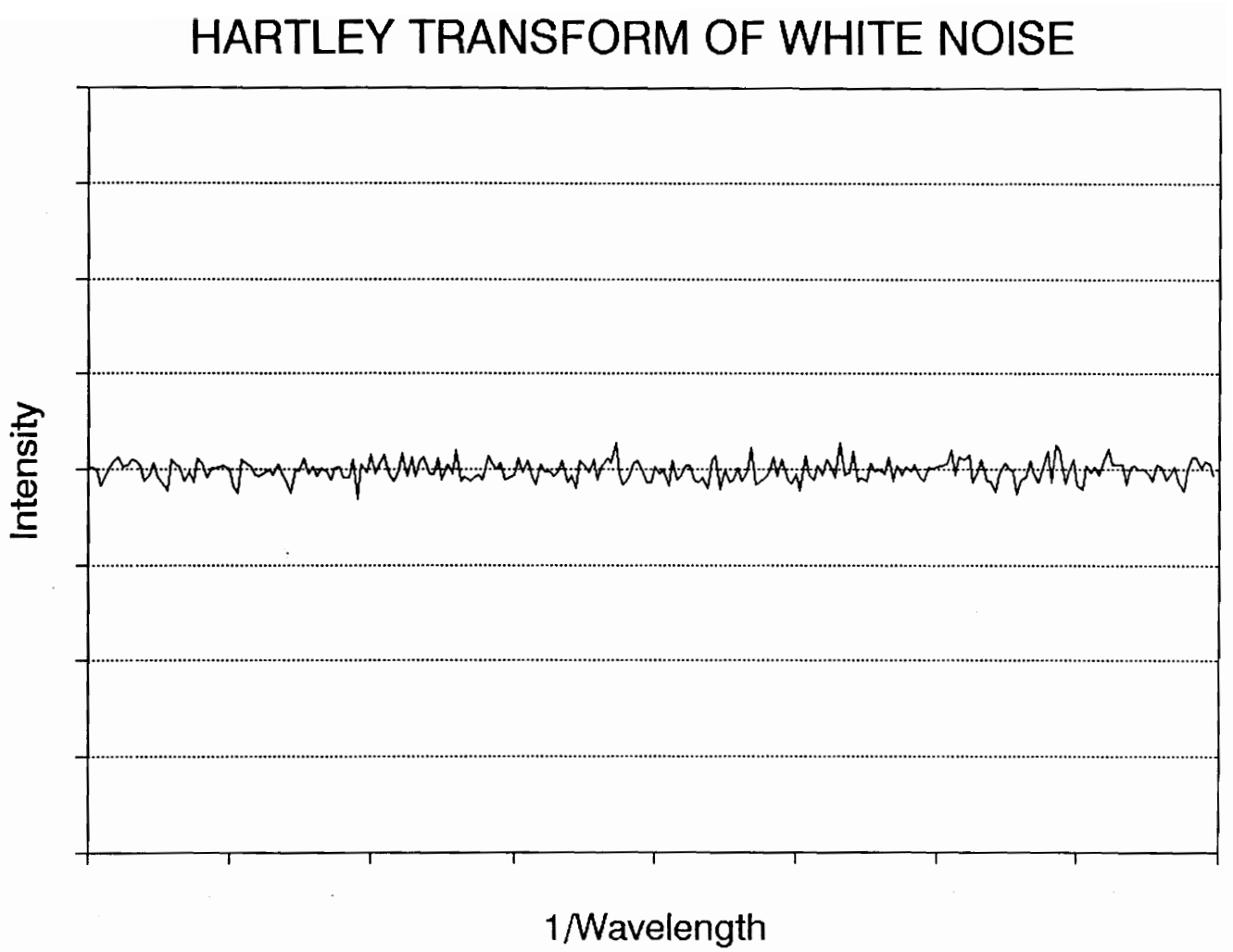


Figure 10. Hartley transform of white noise.

data set.

The above two examples illustrate an important advantage that the Hartley transform domain has over the original spectrum domain for the representation of spectral data. Since the signal in the transform domain is shifted towards the low spatial frequencies and the noise remains spread over the entire spectrum, but with reduced magnitude, the signal to noise ratio, S/N, for points representing low spatial frequencies will be greatly enhanced. Further details on the behavior of white noise and S/N considerations on transform techniques can be found in the references.[71]

The way that the algorithm, which will be explained in detail in a later section, takes advantage of the above discussed properties of the Hartley transform will fully be realized in the following sections.

C. DATA COMPRESSION - LIBRARY SEARCH

The computer industry has made much progress over the past decade in the development of powerful and inexpensive microcomputers. These advances have led to the wide availability of computer controlled instruments capable of generating large quantities of data, and data acquisition systems which collect and store the data. Modern instruments are often equipped with computers that include

many megabytes of disk storage space.

As is often the case, though, even those massive storage devices come to a point where they are no longer adequate for storing all available experimental data. Multidimensional NMR spectroscopy is such an example. Current 2-D spectra are typically contained within matrices representing 4Kx4K or 8Kx8K data points, which require several megabytes of disk storage space.[72] Future 3-D and 4-D data sets, clearly will require much more.

Full resolution - full intensity spectra contain the maximum amount of system information, but occupy the largest amount of computer storage space. Prior to inclusion in a reference library those spectra should be preprocessed in order to reduce the storage requirements, but also to increase the speed of the library search. For that reason several preprocessing - data compression methods have been developed. In this work two of those methods will be used, and will be discussed further.

As was demonstrated earlier, when a spectrum with broad, rather featureless peaks is Hartley transformed, the great majority of the signal is shifted toward the low spatial frequencies, leaving high spatial frequencies vacant of signal, containing only noise. Thus, if only points representing low spatial frequencies in the Hartley domain are stored, significant amount of storage space can be

recovered without any significant loss of system information.[60]

The compression factors achieved with this method are not great, usually less than 10, but the information loss is almost zero, thus making the method very attractive. The original spectra can always be regenerated by inverse transforming the stored data array, after filling zeros in the positions of the points which were not saved.

In systems where the regeneration of the original spectra is not necessary, even some of the low spatial frequencies containing signal can be discarded, as long as the remaining points provide adequate distinction among the spectra in the reference library.

The second compression method that will be used in this work, and which can achieve much greater compression factors, is called spectral encoding. Encoding is a technique where commonly appearing patterns in the original data are replaced by a unique combination of symbols, usually 1's and 0's. The choice of symbols to use is not important, the only requirement is that the symbols should take less storage space than the patterns in the original data that they replace. Choosing 1's and 0's as the symbols to use is very suitable, since these are the two smallest pieces of information computers can store (bit), and at the

same time computers can process them with great speeds.

Encoding methods are very common in other areas where data compression is desired. Text files, for example, are often compressed using encoding methods.[73] Combinations of two, three or even more letters that are often found in words, are replaced by unique combinations of 1's and 0's. The compression factors achieved in those cases vary, depending on which letter combinations are chosen to be substituted, and how often they appear in the particular text file.

In spectral encoding methods, often called clipping, intensity information is converted into 1's and 0's, depending on the magnitude of the intensity. Each point is compared against a predetermined threshold value; if the intensity of the point is above the threshold value the point is replaced by a 1; if the intensity of the point is below, it is replaced by a 0. This method reduces peak information into two levels, binary encoding. A very commonly used threshold value for this method is zero, producing the zero-crossing clipping algorithm.

It is also possible to clip intensity information into more than two levels, by using more than two symbols and more than one threshold value. Three level encoding, trinary encoding, requires three symbols and two threshold values, four level encoding, tetranary encoding, requires

four symbols and three threshold values, etc. For example, in trinary encoding, 0, 1, and 2, can be used as the three symbols, indicating no intensity, small intensity, and large intensity, respectively. In those methods, points in the original spectrum, or the transformed spectrum, are encoded depending on their absolute intensities.

Relative encoding methods, where the points are encoded depending on their relative intensity to surrounding points, also exist. In those methods, the intensity of each point is not compared against a universal threshold value, but a new threshold value is calculated for each point based on the intensity of the surrounding points. Of course those methods require more calculations and longer times to be completed, but they have the advantage of reflecting more fine structures, making possible the distinction between similar peaks.

More details on the zero-crossing clipping algorithm, along with three relative encoding methods, one binary and two trinary, which will be tested for their value in the present algorithm, will be given in later sections.

In library searching techniques an unknown spectrum is compared to each member of a reference library. Reference spectra are sorted in order of decreasing similarity, and a 'hit-list' of spectra which best match the unknown is generated. Spectra in the reference library are high-

quality and usually are stored in a compressed form, which saves disk storage space. At the same time this makes the search task easier and faster. Libraries which compress the spectra in a form that does not allow the search to be carried out in the compressed form, but require the spectra to be regenerated prior to the search, are not very attractive due to the long search times that are required. The main reason for the long search times is the regeneration process.

There are some other important parameters, besides reference spectra storage format, which have to be considered when designing a library search system. One such parameter is the way in which the unknown and reference spectra will be compared. The 'comparison metric' can be based on similarities or dissimilarities between the unknown and the reference spectra and the 'comparison metric' could even weight selected regions of the spectrum differently from others to achieve better discriminations between spectra.

Another parameter that needs consideration prior the development of a library search system is a compromise between the time required to do the search, speed, and the quality of the results expected from the search, performance. Usually, library systems developed with compression algorithm which achieve high compression

factors, have short search times, high speed, but display poor performance. On the other hand, libraries where low compression factors are used for the storage of the reference spectra take longer to search but can achieve higher performance. A compromise is inevitable.

Library search methods, based on the way the search is performed, are divided into two large categories: forward search methods and reverse search methods. In forward search methods the characteristics of the unknown spectrum are compared against those of the reference spectra. On the top of the 'hit-list' will be placed the reference spectrum whose characteristics best match these of the unknown. In reverse search methods the characteristics of each reference spectrum are compared against those of the unknown spectrum. In this case, if all the characteristics of the reference spectrum match those of the unknown, even if some characteristics of the unknown remain unmatched, the reference is a good match and will be places on the top of the 'hit-list'.

On forward search systems the interest is on the best match. On reverse search systems the interest is on the subset of spectra which will best match the unknown. The main advantage of the reverse search is that it can be used for the analysis not only of pure compounds where an exact match is expected, but also in the analysis of mixtures.

Obviously, since the present work focuses on the analysis of mixtures, a reverse search library system will be used. The 'comparison metric' used is based on the similarity of the encoded patterns of the lower spatial frequencies of the Hartley transform of the spectra.

D. NON-NEGATIVE LEAST-SQUARES

The method of least-squares was first proposed as an algebraic procedure by Legendre in 1805, and later justified as a statistical procedure by Gauss in 1809. The technique was adopted almost immediately as the standard procedure for the analysis of astronomical data. Over the years it has spread to all fields of science and now it is one of the most familiar and most widely used multivariate statistical procedures.

A definition of the least-squares problem would be appropriate at this point. Although a strictly mathematical definition of the problem can be given,[74] it would probably be confusing. Instead, an explanation of the problem will be attempted, through the most simple of the least-squares methods, linear regression.

The simplest type of model relating the response, y, to an independent variable, x, is the equation of a straight line:

$$y=\beta_1 x+\beta_0 \quad (32)$$

where β_0 is the y-intercept (value of y when x=0) and β_1 is the slope of the straight line. The plot in Figure 11, shows a set of (x,y) pairs, where it can be seen that a straight line would adequately describe the trend in the data. If an attempt is made to use a ruler to draw a straight line over these points, each time a different line will be draw. An objective method, which will find the straight line which most accurately describes the linear trend of the data is required. Several such methods are available, each one using a different criterion to select the best line. The most commonly used one is linear regression.

Letting $\hat{y}$ denote the predicted value of y for a given value of x, then the error of prediction, e, often called residual, is $e=(y-\hat{y})$, the difference between the actual value of y and the predicted value, Figure 12. Thus, the equation that would accurately describe the sample points can be written as:

$$y=\beta_0+\beta_1 x \quad (33)$$

The criterion that the method of linear regression employs to estimate the y-intercept and slope of the regression line, i.e. choose the best prediction line, is

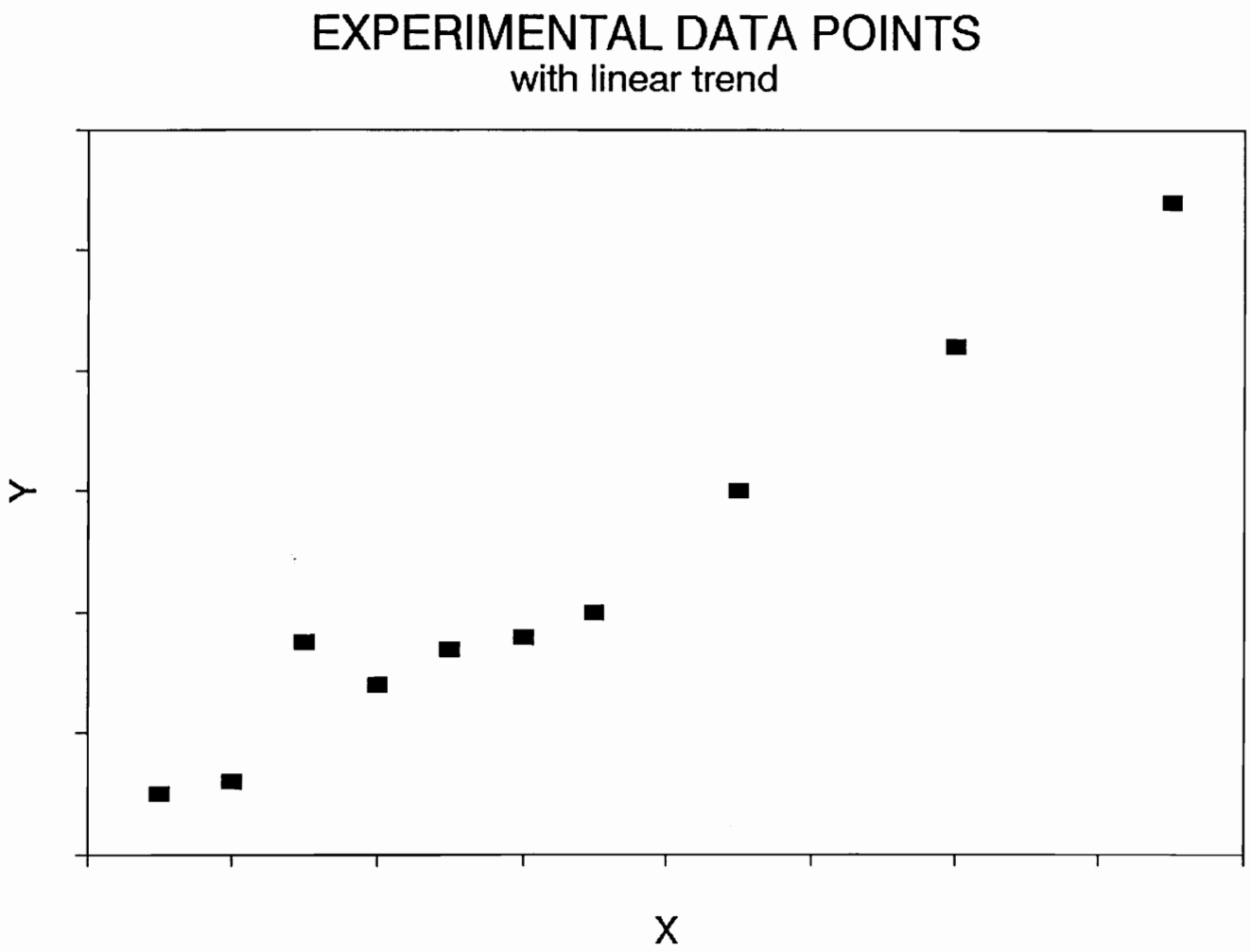


Figure 11. Linear regression points.

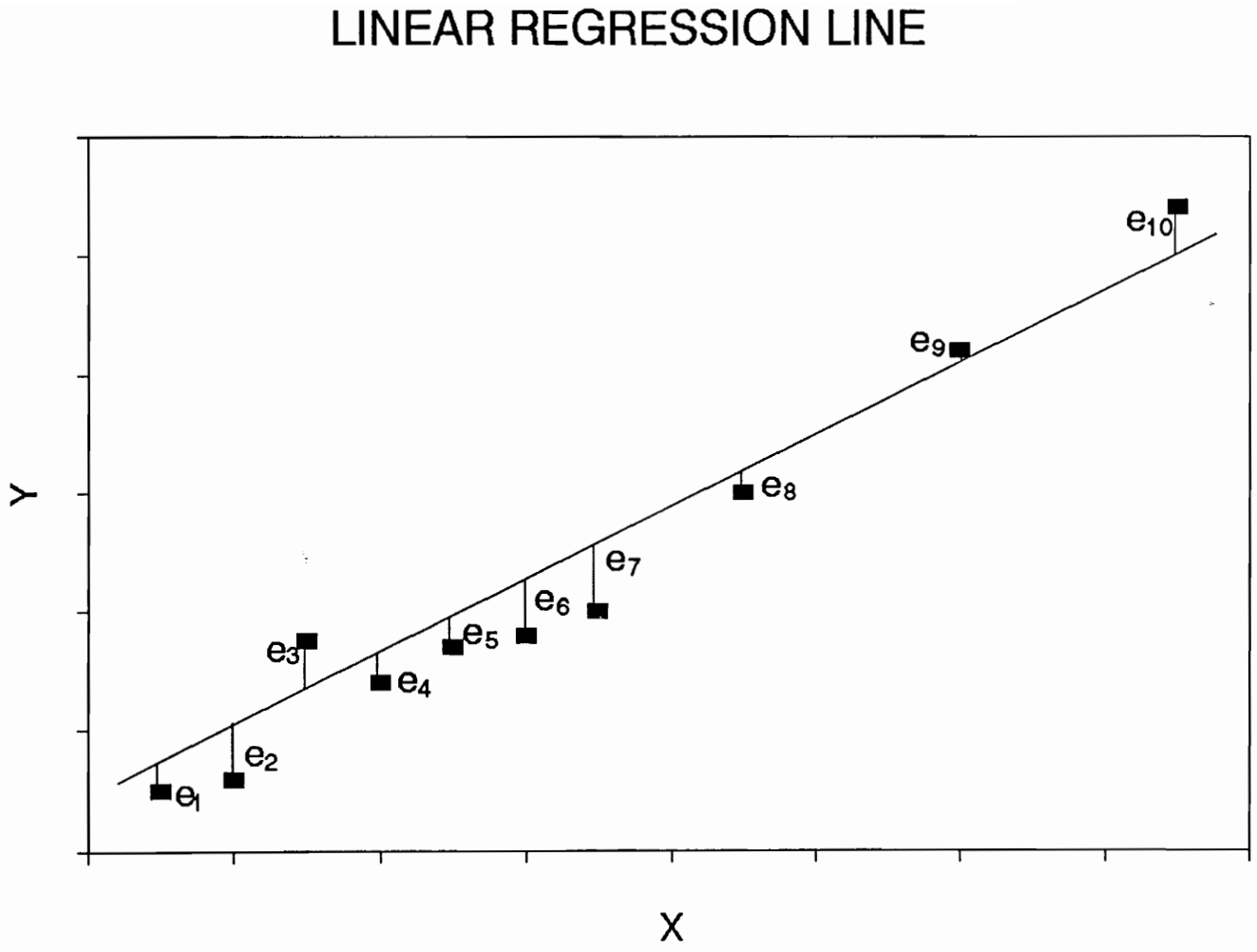


Figure 12. Linear regression line.

the sum of the squared errors of prediction for all sample points:

$$\sum e = \sum (y - \hat{y})^2 \qquad (34)$$

The line which will minimize the sum of the squared errors (least-squares) is the one which will be chosen to most accurately describe the sample points. Further explanation of the calculations required to achieve this goal is beyond the scope of this discussion. Details can be found in the literature.[75]

The same method and criterion can be used in the case of EEMs. Equation (20) was shown to describe the Excitation-Emission Matrix of an unknown mixture as a linear combination of the component matrices. Rewriting that equation for the case of a two component mixture and also including the error of prediction term, Equation (20) becomes:

$$M = \beta_1 M_1 + \beta_2 M_2 + e \qquad (35)$$

The similarity of this equation with Equation (33) is apparent.

In the case of the EEMs, the criterion of the sum of the squared errors of prediction can be used to estimate the values of the β_1 and β_2 concentration parameters, which will best describe the points in the unknown matrix.

Several mathematical procedures are available to carry out the above calculations.[76] The description of those procedures is also beyond the scope of this discussion, but one issue has to be clarified: these procedures are statistical procedures and as such they require a large number of data points for the calculated estimations to be accurate. That greatly increases the number of required calculations, but the power these procedures offer justifies the cost. Also, with the great speeds of new computers the cost is minimal.

There are many applications in applied mathematics, physics, chemistry, statistics, economics, and other fields, where the use of the least-squares method as explained above is not adequate. Usually, some additional information is available about the problem on hand which has to be considered when the problem is formulated. This additional information can be included into the problem by the introduction of certain equality or inequality constraints.

For example, the addition of constraints gives the ability to consider least-squares problems where each variable independently is bound between a lower and upper value, or that the sum of all variables does not exceed a specified value, or that all variables are non-negative. The last constraint generates the least-squares approach commonly known as Non-Negative Least-Squares, or NNLS.

More specifically, the NNLS method can be used in the case of EEMs to solve the least-square problem in Equation (33), imposing the constraint that all the calculated concentration coefficients, β_i, are non-negative. The non-negativity constraint, in this case, includes additional knowledge about the system, that the concentration of a compound in a sample can be positive or zero, i.e. the compound is present in the sample or it is not.

In the present work, the method of least-squares will be used to estimate the concentrations of compounds in a mixture, but at the same time identify compounds which are not present in the mixture. Following the above discussion, the NNLS algorithm will be the most appropriate to use. A FORTRAN implementation of the NNLS algorithm given by C.L. Lawson and R.J. Hanson[74] will be used to carry the calculations.

IV. ALGORITHM

Before the development of computer based databases, spectroscopic data were compiled in books and journals. Each spectrum was stored and printed just as it was collected with no modifications. In the case of a computer database, though, before the data are stored, a number of decisions have to be made and a number of questions have to be answered. These questions concern (a) what data are to be saved, (b) in which format should this data be stored, and (c) how information will be retrieved from the library.

These important aspects of a computer oriented database will be address in this chapter. First, the philosophy behind the storage/search algorithm will be discussed. Then, the procedure followed to develop a spectral library will be explained. Finally, the complete algorithm will be shown in the process of analyzing an unknown mixture.

A. ALGORITHM PHILOSOPHY

The single most important consideration prior the development of any computer application is the actual methodology which will be used to achieve the desired result. In the presented case, the questions of how the algorithm will identify the components of a mixture, and how it will quantify them are presented.

Although, these questions appear to be very simple, the

actual realization of an appropriate algorithm is difficult. An explanation of the methodology used to analyze, qualitatively and quantitatively, three dimensional fluorescence spectra of unknown mixtures follows. This methodology is schematically summarized in Figure 13, and its architecture is developed in the following paragraph.

Assume that a spectral library, containing the EEMs of pure compounds, has been developed, and the spectrum of an unknown mixture has been collected. The algorithm filters the members of the library, leaving to pass through only those which are most likely to be components of the mixture. The parameters of the filter, i.e. which library members will be allowed to pass, are controlled by the unknown EEM. The final estimation of the number and identity of the components, as well as their concentrations in the mixture will be made by a Least-Squares method. Only the Hartley transforms of the EEMs of those compounds which passed through the filter will be used in the Least-Squares method.

As was explained in the library search theory section, a compromise between speed and accuracy in the prediction during a library search has to be made. The approach chosen in the present work was to select speed over accuracy. The filter is not expected to be totally accurate. It is only expected to filter out the majority of the library members, passing through the actual components of the mixture plus a

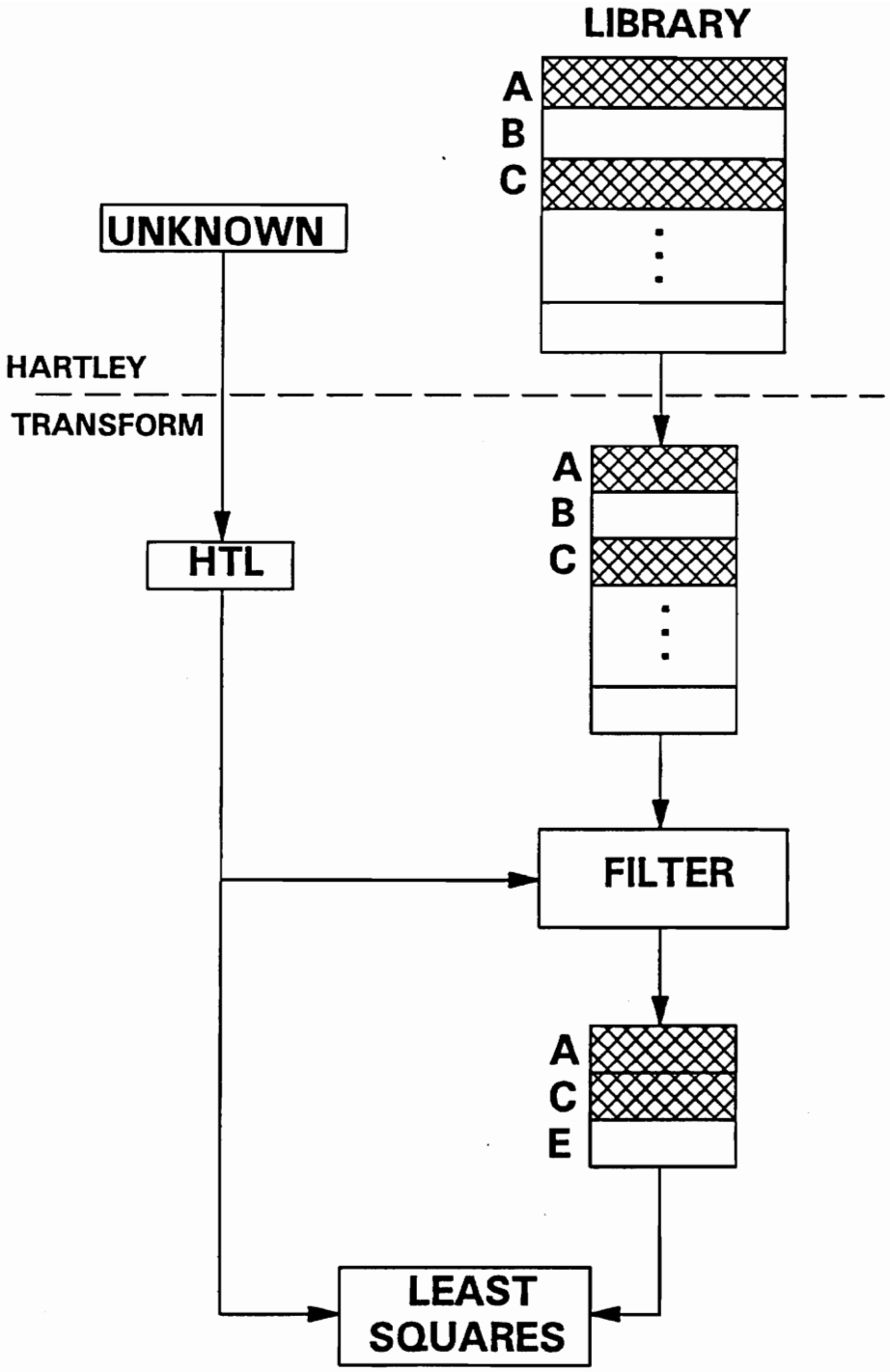


Figure 13. Philosophy of the presented algorithm.

population of spectra which at a first approximation might be possible components. Too tight a filter at this stage would be detrimental.

In cases of very complicated unknown mixtures with a large number of components it is expected that members of the library which are actual components maybe filtered out during the filtering step. The way the algorithm deals with such a situation will be explained in later sections.

B. LIBRARY DEVELOPMENT

Which data are needed and therefore stored on the disk depends on the purpose the data is to serve. In the case of the algorithm under investigation, the goal is both qualitative and quantitative analysis of mixtures. Therefore, the stored data must contain information which will allow to distinguish between members of the library, i.e. qualitative information, and also provide means for concentration computations, i.e. quantitative information.

The approach chosen in this work was to divide the stored data into two separate parts, each containing a different type of needed information. One part contains only enough information to allow the algorithm to identify the members of the library. The other part verifies this initial identification and gives concentration estimation for the components.

This scheme for the storage of data in the spectral library, along with the steps required to achieve such information separation, are shown in Figure 14. The number of data points involved in each step of the process is shown in parenthesis. In the case of the CLP files, the number in the parenthesis denotes the number of clipped points, or the length of the clipped pattern.

A description of the process for the development of the spectral library follows, along with an explanation of the purpose each step serves. The process must be applied to each and every EEM that will be a member of the library.

The first step toward the development of the spectral library is the unfolding of the 64x64 points, three dimensional spectrum, into a linear array of 4096 points, as shown in Figure 15.

The reduction of the dimensionality of the spectrum accomplishes two goals. First, it speeds up the next step of the library development process, which is the transformation of the data using the Hartley transform. By going from a three dimensional data structure, which would require a two dimensional transformation, to a linear array, which requires a one dimension transformation, there is a reduction in the number of calculations required to transform the data.

The theory of the Hartley transformation at first

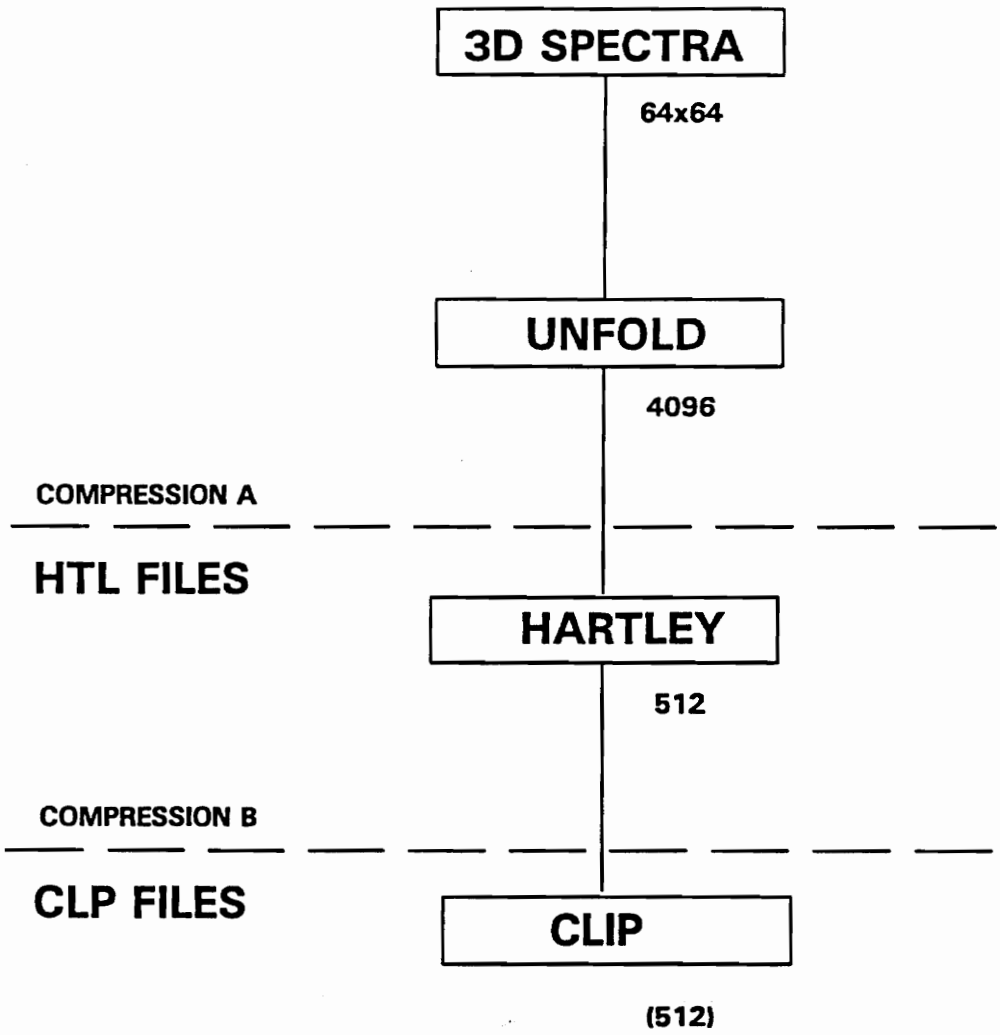


Figure 14. Library development scheme.

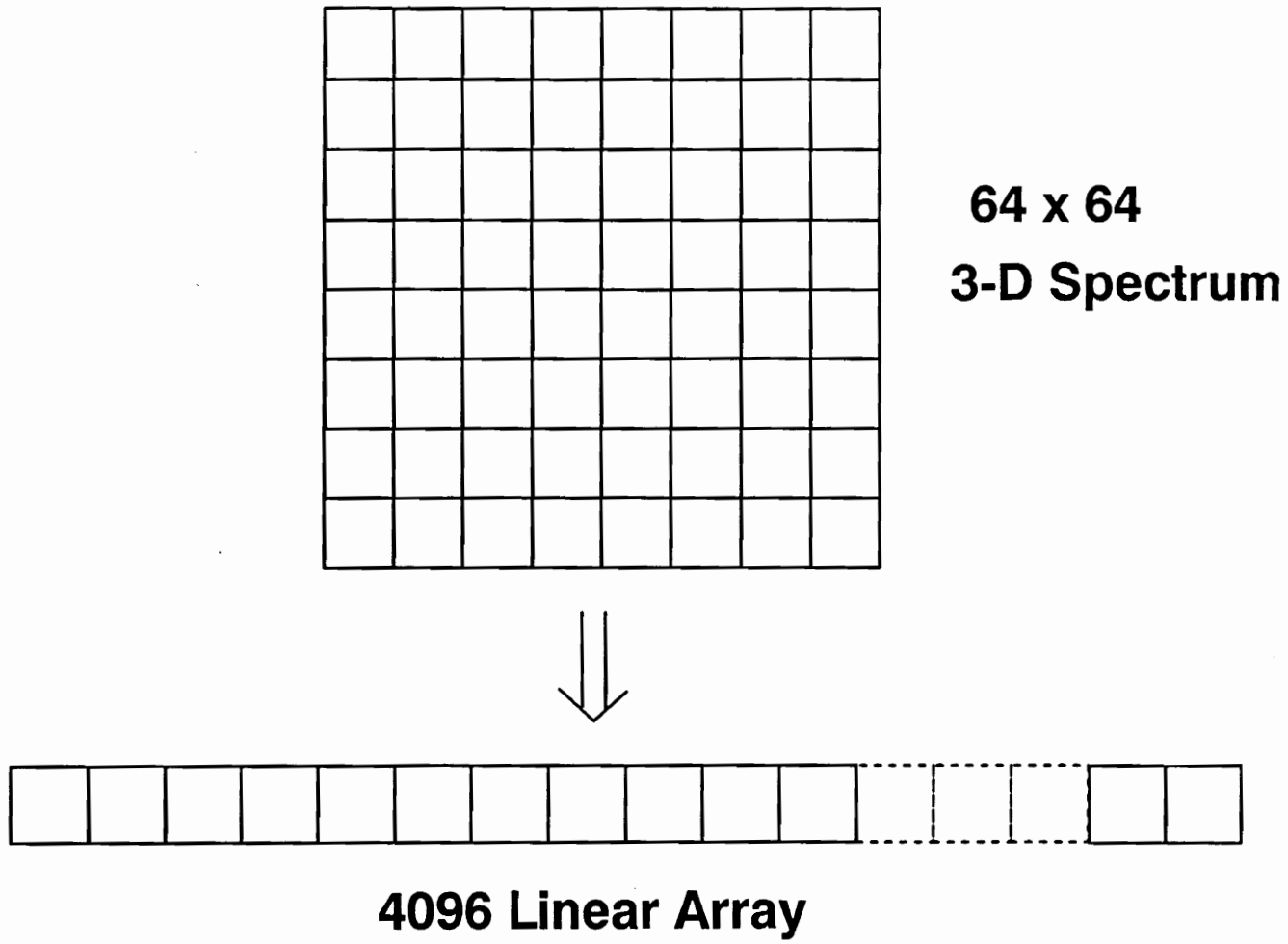


Figure 15. Unfolding of a spectrum into a linear array.

glance might not support the above argument : since the number of points is the same in both case, the number of calculations should be the same. However, what must be considered, is that the transformation is carried out in a computer program. There are significantly more instructions, mainly I/O instructions, that have to be carried out in the case of the two dimensional transformation than in the one dimensional transformation.

In the present work all the computations were carried out on a very fast computer (DEC VAX), and the speed issue might be considered academic. For a PC implementation of the algorithm with larger data sets the difference would become significant.

The second, and more important goal that the reduction of the dimensionality accomplishes is to increase the probability of identifying the members of the library within the spectrum of an unknown mixture. The next step in the development of the library is the transformation of the spectrum. In the case of a one dimension transformation the signal will be more evenly spread over a larger number of points in the transform domain (see Figure 16). In the case of a two dimension transformation the signal would be forced into a smaller number of points corresponding to few low frequencies (see Figure 17).

The end result of unfolding is that the patterns which

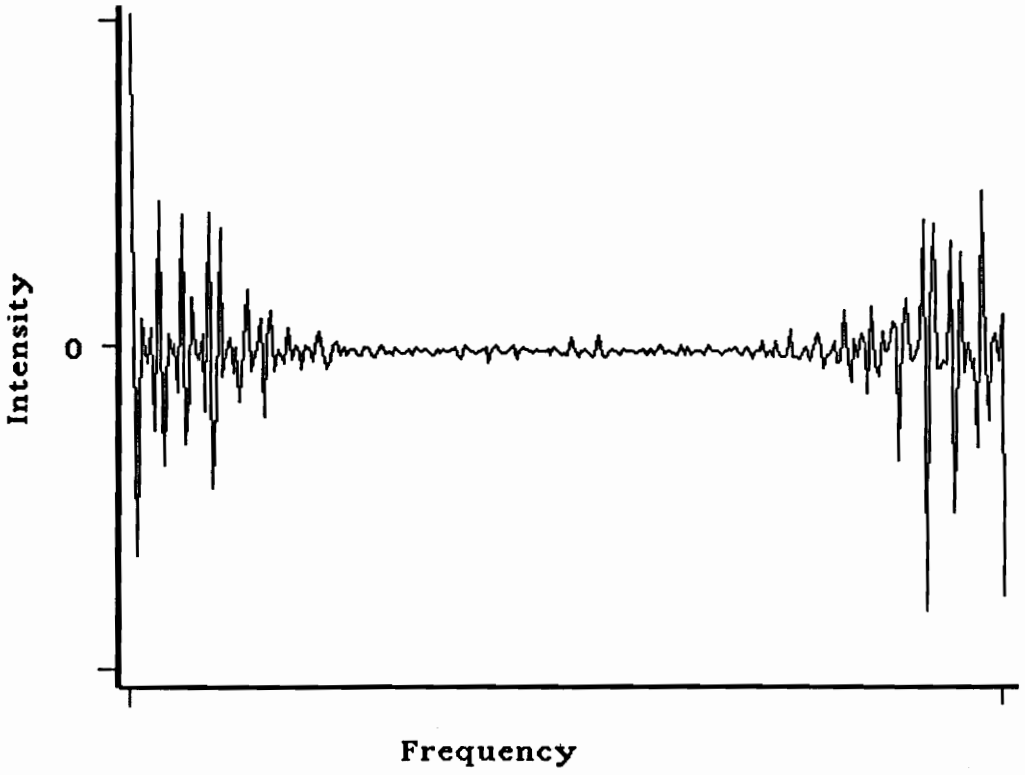


Figure 16. Typical transformed unfolded spectrum.

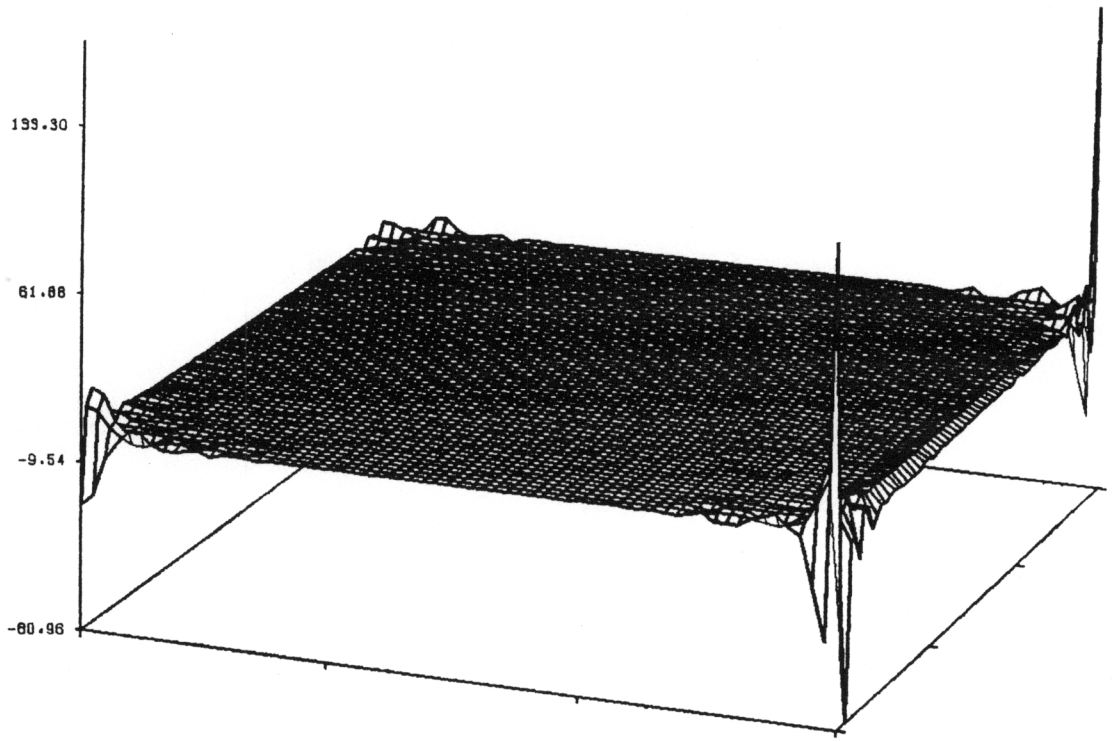


Figure 17. Typical two-dimensional transformed spectrum.

emerge from the spectral encoding or clipping step, which follows the transformation step, will be longer and thus more distinguishable from each other. Consequently, during the reverse library search the identification of the components in a mixture will be possible, even in cases involving significant spectral overlap of the components in the mixture.

The next step toward the development of the spectral library is the transformation of the spectra using the discrete Hartley transform. A typical spectral transform on an unfolded data set is shown in Figure 16. It can be seen that the signal has been moved to the two ends of the transform domain, which correspond to low spatial frequency components. The middle of the transformation, which correspond to high spatial frequency components, is virtually free of signal. As was explained in the section on the Hartley transform theory, the high frequencies contain only components which correspond to white noise present in the spectrum.

The transformation of the spectrum during the development of the spectral library accomplishes a three-fold goal. First, it transforms the spectrum to a form with much more characteristic than the original one, making possible the quick and accurate identification of the individual spectra in a mixture.

The fluorescence spectra and Excitation Emission Matrices of pure compounds are rather broad and featureless, with not a lot of sharp characteristics. This makes visual distinction between two spectra very difficult. In the case of mixtures involving overlapping peaks the task of distinguishing the spectra of the individual components becomes impossible.

After the transformation the spectrum contains a large number of narrow, well defined, positive and negative peaks, which can very easily be compared against those of an unknown mixture to determine the identity of the compounds present in the mixture.

The second goal that the transformation of the spectrum accomplishes is to minimize the amount of space required to store the spectrum in the computer memory. As was explained in previous sections, the majority of the signal is shifted toward the low spatial frequencies, leaving high spatial frequencies vacant of signal, containing only noise. Thus, if only points representing low spatial frequencies in the Hartley domain are stored, significant amount of storage space will be preserved. The original spectrum can always be regenerated with those few points which have been saved.

Furthermore, since in terms of the present algorithm the only critirion is to save enough information so that the identity and quantity of the different compounds in the

mixtures can be estiblished and the regeneration of the original spectrum is not required, the number of points stored can further be reduced by discarding some low spatial frequencies. In fact, half of the low spatial frequencies can be discarded. Only the first 512 points from one side of the transform domain were saved, and stored as part of the library. This is called truncation. Those 512 points, for each member in the library, were stored in files as four byte integers, with file extension HTL. These files will henceforth be referred to as HTL files.

The third but very important goal the transformation of the spectrum accomplishes is to improve the performance of the Non-Negative Least-Squares method which will be used to estimate the concentrations of the compounds in the unknown mixture. The NNLS method will use the points stored in the HTL files for the calculation of the concentrations.

Although, the method of Least-Squares is very powerful and very robust, it can produce erroneous results when used with points with a low signal-to-noise ratio, S/N, or when some of the points used do not contain any signal, but only noise. For example, in a case of two component mixture where the two components give signal in two separete spectral regions, it is important that points from both regions be used if the method of Least Squares is to be used for the estimation of the concentration of the components.

If points only from one spectral region are used the method will produce erroneous results.

The transformation along with the truncation of the spectra guarantees that the above two deleterious conditions do not exist. First, the transformation, as was explained earlier, increases the signal-to-noise ratio for the low spatial frequencies in the transform domain; thus only points with high signal-to-noise ratio will be used for the Least Squares method. Second, since after the truncation only those low spatial frequencies are stored in the HTL files all of the points used in the Least Squares method will contain an optimum signal level and points containing only noise will never be used.

Finally, the third and last step toward the development of the spectral library is the spectral encoding or clipping of the points saved in the HTL files. In a first attempt the zero-crossing clipping algorithm was used to reduce the points in the HTL files into a 512 points long combination of 1's and 0's, producing a pattern unique for each member of the library. Each of these combinations was stored, into a file with the extension CLP. These files, referred to as CLP files, along with the corresponding HTL files, are what make up the spectral library which will be used by the algorithm.

In the clipping step the intensity information, which

was preserved in the HTL files after the transformation, is now lost. The CLP files contain only the qualitative information for each spectrum in the library, leaving the intensity or quantitative information in the HTL files. The CLP and HTL files are the two separate representations of the data, containing two different types of information, of a qualitative and quantitative nature respectively.

The clipping of the data has a tremendous impact on the storage space required. With the zero-crossing clipping algorithm, only one bit per point is required to store the clipped pattern. The other three relative encoding methods, which were also tested in this work, require three symbols to be used for the encoding (-1, 0, and 1). Two bits per point are required to store the clipped pattern.

The compression that is achieved by discarding most of the spatial frequencies after the Hartley transformation of the spectra, Compression A, along with the compression achieved with the spectral encoding, Compression B, are summarized in Figure 18.

The original spectra contain 4096 points. Each point takes four bytes if stored as a long integer. Each original spectrum requires 16,384 bytes. After all the high spatial frequencies, and half of the low frequencies are discarded only 512 points of the Hartley transformation need to be stored in the HTL files (Compression A), again as four

COMPRESSION

3D SPECTRUM (64 x 64)

4096 points x 4 bytes/point = 16384 bytes

COMPRESSION A

512 points x 4 bytes/point = 2048 bytes

(8:1 or 12.5%)

COMPRESSION B

512 points x 0.25 bytes/point = 128 bytes

(128:1 or 0.8%)

Figure 18. Compression achieved during the library development.

byte long integers. The space each spectrum requires is now reduced to 2048 bytes. This is a compression ratio of 8:1. Only 12.5% of the storage space occupied by the original spectrum would required.

After the HTL files are clipped, each of the 512 points requires at most (in the case of the relative encoding methods) two bits or 0.25 bytes. Thus, the whole clipped pattern to be stored in the CLP file is only 128 bytes long. This is a compression ratio of 128:1. The CLP files occupy less than 1% (0.8%) of the original spectral space.

The fact that only the qualitative information stored in the CLP files will be used during the library search tremendously reduces the time necessary to execute the search algorithm. The library search is the time critical part of any algorithmic method since it is usually done in user relevant time.

The overall compression ratio achieved during the development of the library, since both the HTL and CLP files would have to be stored in the library, is about 7.5:1, requiring only about 13% of the original space required to store spectra in their original form.

One point need clarification. All the spectra have to be collected under the same conditions. These conditions include the excitation and emission wavelength ranges, the resolution of the spectrum, the width of the slits of the

excitation and emission monochromators, the signal amplifier gain, etc. The instrument parameters used for the development of the spectral library used in this work will be given in the experimental section.

A description of the complete algorithm now follows. The description will be given from the point of an unknown mixture, i.e. the procedure which has to be followed in order to analyze an unknown mixture.

C. ANALYSIS OF A MIXTURE

The discussion in this chapter assumes that the steps previously explained in the development of the spectral library have been completed, the HTL and the CLP files for all the compounds in the library have been stored, and the three dimensional fluorescence spectrum of the unknown mixture has been collected. The spectrum of the unknown mixture must have been collected under the same conditions used for the spectra in the library.

The first four steps in the process of analyzing the spectrum of an unknown mixture are the same followed during the development of a compressed spectral library component. The unknown three dimensional spectrum is unfolded into a linear array, transformed using the Hartley transform, the first 512 low spatial frequencies are stored in an HTL file, these same 512 points are clipped, and saved in an CLP file.

The complete algorithm, including these four initial steps, is show diagrammatically in Figure 19.

After the creation of the HTL and CLP files of the unknown, the reverse search of the library using only the information in the CLP files follows. The goal of this search is to eliminate the majority of the library members on the basis of their improbability of being components of the unknown mixture. This is done by checking the similarity of the encoded patterns of each library member against the encoded pattern of the unknown spectrum. The greater the similarity of the encoded patterns, the greater the probability that the reference compound is a component of the mixture.

As was explained in the theory section, the test of the similarity of the encoded patterns is done on a bit basis. In particular, if a certain bit of the encoded pattern of the reference spectrum matches that of the unknown, this point is considered to be a positive attribute. If it does not match it is considered a negative attribute. The "positive" to "negative" attribute ratio, Positive/Negative, is a measure of the similarity of the two encoded patterns, and is the comparison metric used by the algorithm for the reverse library search.

In the case of two totally unrelated spectra, the number of "positive" points is expected to be found equal to

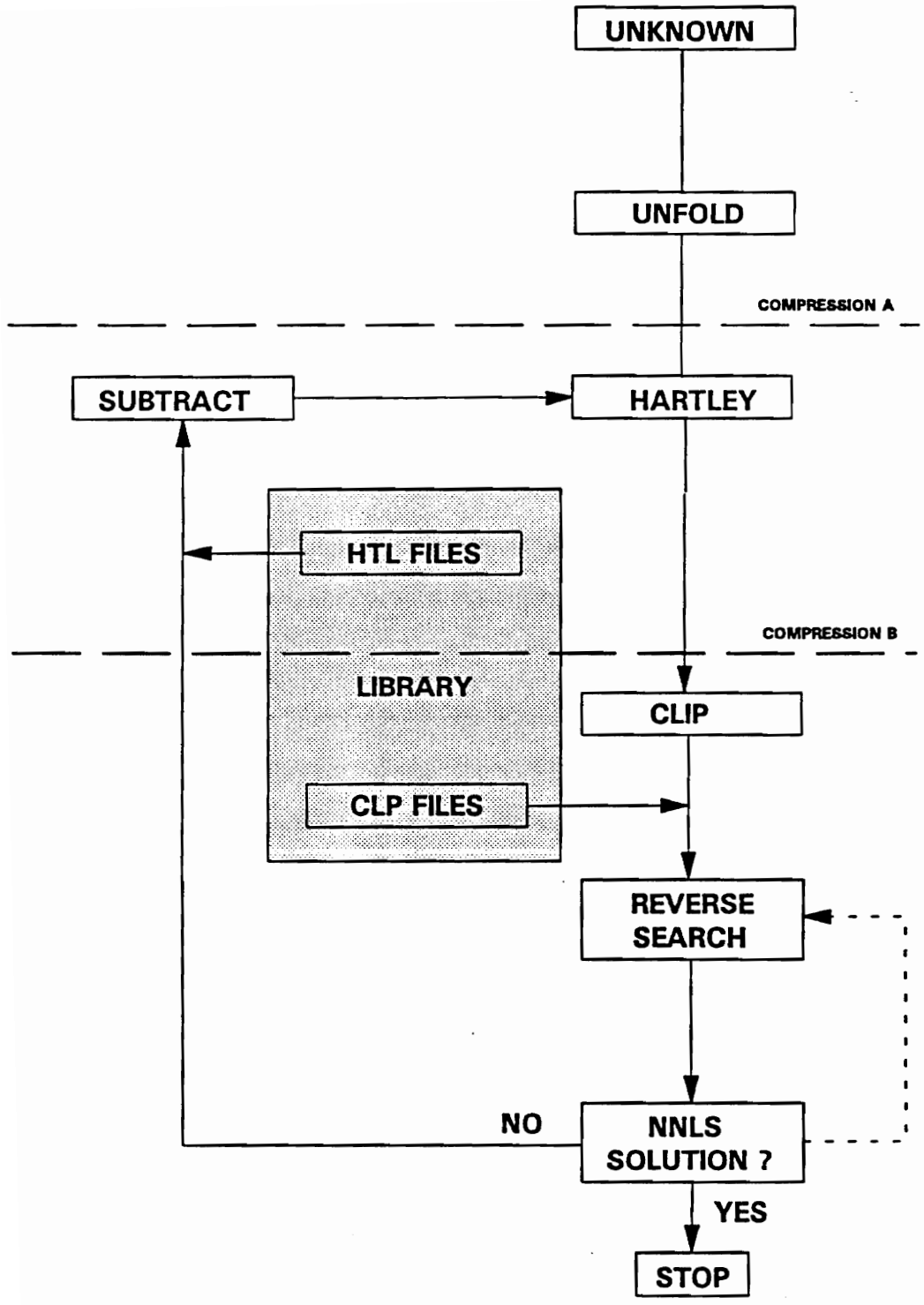


Figure 19. The complete algorithm.

the number of "negative" points, giving a Positive/Negative Ratio equal to 1.0. In the case of a reference spectrum which is an actual component of the mixture, a much larger number of "positive" points compared to "negative" points is expected, giving a high value of Positive/Negative Ratio. The higher the Positive/Negative Ratio, the higher the similarity of the encoded patterns, which indicates that there is a high probability the reference compound to be a component of the mixture.

In the case of a trinary encoding method, since there are three symbols used, -1's, 0's, and 1's, there would also be non-applicable, N/A, points. Those would be combinations for a particular bit pair between that in the reference spectrum and that in the unknown 1,0; -1,0; 0,1; or 0,-1.

The reason such points would not be used, and thus are termed non-applicable, is because of the presence of white noise. As explained in the theory of the Hartley transform, white noise from the original spectrum, will be evenly distributed in the transform domain over the entire spectrum. The presence of this noise can force a point to be moved from a 0 to a position of 1 or -1, or vise versa. Thus combinations involving 0's cannot be of any value. In the case of binary encoding methods, since only two symbols are used, 1's and 0's, such distinction is not possible.

During the library search step of the algorithm

(Figure 19), the Positive/Negative Ratio for all the reference spectra is calculated, and the library members are sorted from highest to lowest ratio. At the top of the list are the reference compounds with the highest probability of being components of the unknown mixture. Only those compounds would be used for the next step of the algorithm, which is the method of Non-Negative Least-Squares.

During the NNLS step, the HTL files and the quantitative information contained in those files will be used. Only the members of the library with the highest Positive/Negative Ratios, will be involved in this step. The NNLS method is expected to verify the selection of compounds from the reverse search step. The verification will be achieved by calculating the concentration factors of the reference compounds in the mixture.

For compounds actually present in the mixture, the method would estimate their concentration relative to the concentration of the compound in the reference spectrum. For the members of the library which were selected during the library search but are not actually present in the unknown mixture, the NNLS method is expected to give a concentration estimation of zero.

All Least-Squares methods provide means for the evaluation of the quality of the suggested solution. The implementation of the NNLS method used in the present work

uses the Euclidean length or Euclidean norm, RNORM, defined as :

$$RNORM = (\sum_{i-1}^{n} u_i^2)^{1/2} \qquad (36)$$

where u is the residual vector of the estimated solution. A large value of RNORM denotes a poor estimation, where a small value denotes an accurate estimation.

In term of the present algorithm, a small RNORM value indicates an acceptable estimation of the identities and concentrations of the components of the unknown mixture. At this point the analysis of the mixture has been completed. The number of components in the mixture will equal the number of reference compounds with concentration factors larger than zero. If the estimated concentration factors are multiplied by the concentration of the corresponding compound in its reference spectrum the result will be the absolute concentrations of the components in the mixture.

Where the NNLS method gives rise to a large RNORM value the solution is considered non-acceptable. This would be the result of an incomplete library search, during which one or more components of the mixture were not retrieved from the library. As was mentioned earlier, this is a situation which can arise in cases of mixtures containing a large number of components.

In such situations, the largest estimated concentration factor will be used to subtract the corresponding compound from the mixture. The subtraction will be done in the Hartley domain. More explicitly, the HTL library file of the compound with the highest estimated concentration factor (after each point in the file is multiplied by that factor) will be subtracted from the HTL file of the unknown, Figure 19.

Because of the property of addition of the Hartley transform, explained earlier, the resulting HTL file will be the Hartley transformation of the remaining components of the mixture. This new HTL file will be clipped and further treated as a new unknown mixture for the reverse search of the library.

Again, the Positive/Negative Ratio for all reference spectra will be calculated, and the library members will be sorted from highest to lowest ratio. The compounds with the new higher ratios, as well as the compound subtracted previously, will be used by the NNLS method. The unknown used for this second run of the NNLS method will be the original HTL file of the unknown mixture, not the one found after the subtraction. The reason is that, as was explained earlier, the first estimated concentration for the subtracted component is not expected to be very accurate, thus a new more accurate estimation is needed.

The calculated RNORM will be checked again and either the solution will be accepted, or the algorithm will re-enter the same loop by subtracting a second component. The compound which will be subtracted this time will be the compound with the second highest concentration factor. The algorithm can continue to loop through until a satisfactory solution is found. Of course, to avoid an infinite loop structure in the case a satisfactory solution cannot be found, a limit on the number of times that the algorithm will be allowed to go through the loop must be set.

The choice of subtracting only one compound at a time was made because in the presence of a major component in the mixture it was found that the accuracy of the calculated concentration factors for other components was limited. By subtracting only the component with the highest concentration factor the chance of subtracting a compound not actually present in the mixture was essentially eliminated.

Finally, besides this main loop of the algorithm there is a small branch in the algorithm which is designed to carry information obtained during the course of one loop to the next. As can be seen in Figure 19 there is reverse connection between the step where the NNLS method is applied and the next reverse search of the library. This function is explained in the following paragraph.

It was found that if the concentration factor for a specific member of the library during the NNLS step is calculated to be 0.0, the probability of that compound being a component of the mixture was also zero. Even in the situation where a high Positive/Negative Ratio was calculated for that member of the library during consecutive library searches, that compound would not be further considered in the computations. This information is communicated from one loop to the next through the reverse connection mentioned above.

This concludes the theoretical explanation of the algorithm. In the experimental section which follows a description of how the algorithm actually behaved will be given.

V. EXPERIMENTAL

The study of the behavior of the algorithm with experimental data was carried out in two parts. In the first part, the behavior of the algorithm was tested against computer generated spectroscopic data. In the second part the algorithm was tested against actual Excitation Emission Matrices of multi-component mixtures.

Detailed descriptions of the two parts of the study will follow a brief discussion on the software that was developed to carry out the functions of the algorithm.

A. SOFTWARE

Two programs that would accomplish the two separate functions dictated by the algorithm are needed : the first program processes the EEM of a reference compound and adds it to the reference library. The second program processes the EEM of the unknown mixture and carries out the complete analysis of the mixture. From the discussion in the previews sections it should be obvious that the beginning of the second program would duplicate the actions of the first.

Although this two program approach would be sufficient, in order to facilitate the process of debugging and testing of the code a multi-module approach was employed instead. Each of the steps needed for the development of the reference library, Figure 14, as well as each of the steps

required for the analysis of an unknown mixture, Figure 19, were built as separate modules, i.e. a separate pieces of code.

The computer language chosen for the software implementation and testing of the algorithm was, for the most part, PASCAL, a high level language, widely used in a variety of fields, and available for a great variety of computer platforms.

The presented algorithm involves extensive mathematical manipulations of large data sets, especially during the Hartley transform of the EEMs. It also involves a great number of Input/Output operations during the reverse library search. These functions can easily be carried out by built-in functions and procedures available in PASCAL. To insure the portability of the developed code, standard PASCAL, as it is defined by the American National Standards Institute, (ANSI), was used.

For the Non-Negative Least-Squares part of the algorithm, a FORTRAN implementation of the method, as given by C.L. Lawson and R.J. Hanson,[74] was used.

The computer platform for the development and debugging of the code, as well as the testing of the algorithm, was a DEC VAX/VMS system, (Virtual Address eXtension/Virtual Memory System). At the time the development of the

algorithm was started, the VAX/VMS was the only available system with adequate speed, memory, and disk storage space to handle the requirements of the presented work.

Listings of the code for the major parts of the algorithm, can be found in Appendix A.

B. SIMULATED DATA

1. EXPERIMENTAL

During this part of the study, Excitation Emission Matrices of poly-aromatic hydrocarbons, collected during a previous study that was carried out at the Laboratory Automation and Instrument Design Group of the Chemistry Department at Virginia Polytechnic Institute and State University by Fumiko Ishihara (ref. 59), were selected to develop the reference library. The Excitation Emission Matrices were stored in ASCII format computer files.

Those Excitation Emission Matrices were collected with a Perkin-Elmer Fluorescence Spectrophotometer, Model MPF-66, which provides a 0.25nm to 20nm resolution range in 0.1nm increments (MPF-66 Operating Directions, Perkin-Elmer 1984). The instrument was connected to a Perkin-Elmer 7500 Professional Computer for data collection, and instrument control. The 64x64 data points Excitation Emission Matrices were collected with an instrument resolution of 3nm. More details on the instrument parameters and settings, as well

as the source and purity of the compounds and solvents used can be found in ref. 59.

During this part of the study, simulated EEMs of mixtures were employed to explore different parts of the algorithm. The simulated EEMs of mixtures were developed by mathematically adding the actual spectra of the pure components. To simulate concentration effects, before the spectra of the pure components were added together each pure compound spectrum was multiplied by a concentration factor between 1 and 10. A concentration factor of 1 means that the concentration of the pure compound in the mixture is equal to the concentration of the compound in the reference spectrum, were a concentration factor of 10 means that the concentration of the pure compound in the mixture is 10 times greater than the concentration of the compound in the reference spectrum.

The EEMs of the pure compounds which were added together to develop the unknown mixtures, as well as the concentration factors by which they were multiplied, were selected by a random drawing model. The random drawing was implemented by using a random number genarator. Statistically sound random number generators are part of most computer languages.

In this type of experimental set-up, where the spectra in the reference library are actual EEMs of pure compounds,

but the unknown mixtures are mathematically developed, the behavior of the algorithm is isolated from instrumental noise, quenching, as well as chemical interactions that can excist in the case where EEMs of real mixtures are used. The behavior of the algorithm when those effects are present will be examined in following sections.

1. LIBRARY SEARCH OPTIMIZATION

By examining the diagram of the philosophy of the algorithm, Figure 13, as well as the diagram of the complete algorithm, Figure 19, it should be evident that the filtering or reverse search of the library is the most important step of the algorithm. If during the search all the components of the mixture were retrieved from the library, the NNLS method would readily be able to quantify them.

The effort to optimize the reverse search part of the algorithm i.e. to insure that the algorithm would produce the best possible results at the smallest possible number of iterations through the loop, is the subject of the next section.

In the event where the algorithm can retrieve all the components of a mixture during the first search of the library, the application of the NNLS method should have no problem in accurately identifying and quantifying those

components during the first run. No further iterations through the loop would be necessary.

To achieve optimum results during the reverse library step of the algorithm it is obvious that the process of unfolding and spectral encoding, as well as the number of points used during the reverse library search, should be optimized. Also, with the same goal in mind, an effort to filter the Hartley transform before the spectral encoding step was attempted.

The initial effort was to find the optimum procedure to unfold the three dimensional spectra into a linear array. For that reason two unfolding schemes were investigated. In the first one, the spectrum was unfolded by following a boustrophedon, or zig-zag path, starting from the upper right corner of the spectrum and ending at the lower right corner. The boustrophedon unfolding can be visualized by following the arrows shown in Figure 20.

In the second scheme for the unfolding of the spectrum, a spiral path is followed. Again the unfolding starts at the upper right corner, but this time after moving in circles of decreasing diameters, the unfolding stops at the center of the spectrum. This unfolding procedure will be referred to as spiral unwrapping, to distinguish it from the preview unfolding. Figure 21 diagrammatically shows the spiral unwrapping scheme.

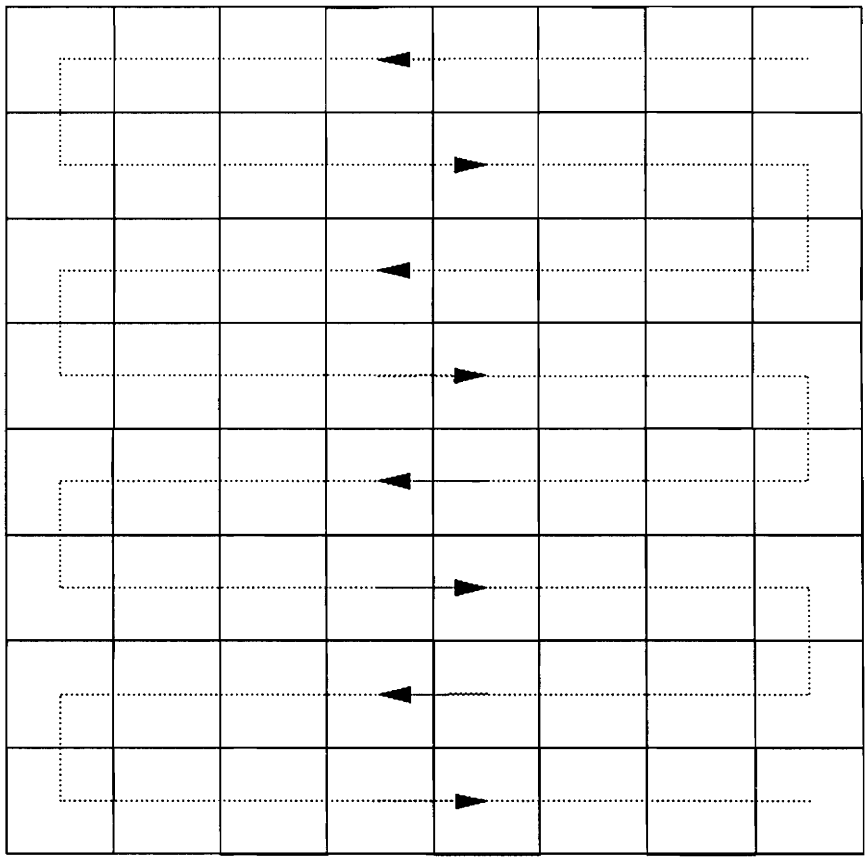


Figure 20. Boustrophedon unfolding of an EEM.

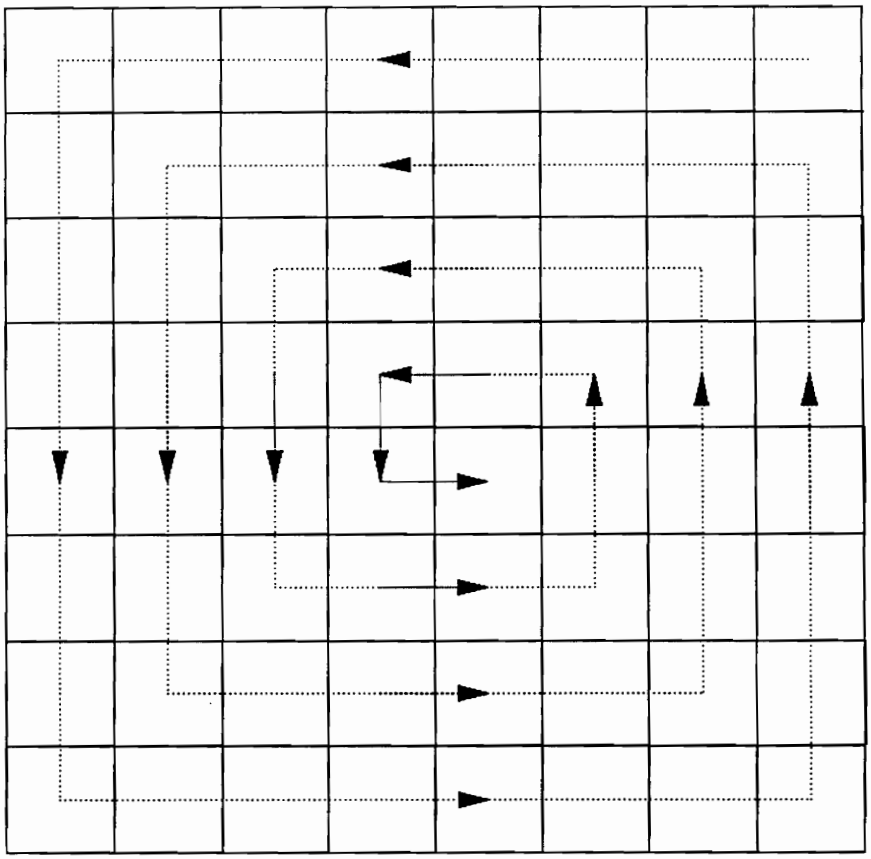


Figure 21. Spiral unwrapping of an EEM.

The other part of the algorithm that was investigated in an effort to optimize the reverse search results was the spectral encoding or clipping step. As was explained in the theory section, in addition to the simple zero-crossing clipping algorithm, three relative encoding methods have been developed and tested for their application in the algorithm. From these three relative methods, the first is a binary method, i.e. it utilizes two symbols to encode the data, whereas the other two are trinary methods, i.e. they utilize three symbols to encode the data.

The three relative encoding methods will be referred to as clipping methods A, B, and C, respectively. These three methods were designed to reflect, progressively from A to C, finer structures in the Hartley transform of the spectra. Unfortunately, each increase in resolution is at the expense of speed. The zero-crossing method, which will be referred to simply as clipping method, is the crudest but at the same time fastest spectral encoding method tested. Figure 22 shows an example of the zero-crossing method.

For the first relative encoding method, clipping method A, two symbols are utilized for the encoding, 1, and 0. The methodology used to achieve the encoding is the comparison of each point with the one immediately proceeding it. The encoding starts with the first point on the left of the transformation, which has always a large positive value, and

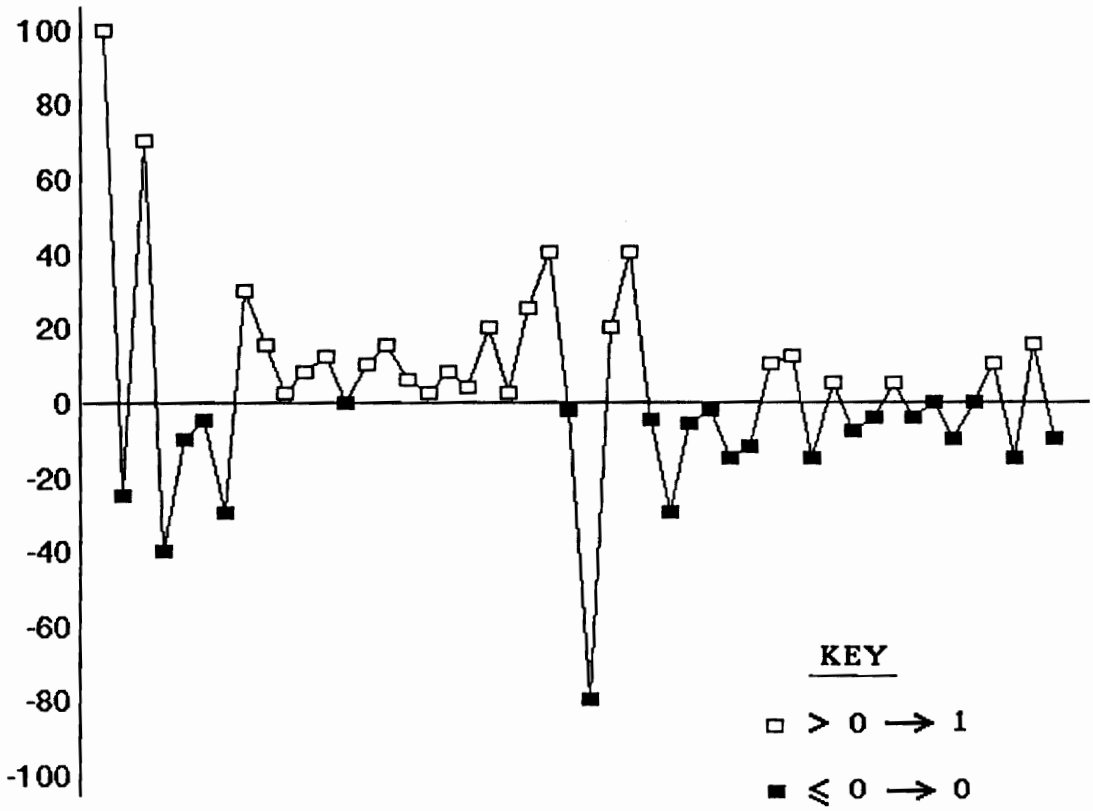


Figure 22. Example of zero-crossing clipping.

thus it is clipped to a 1. This first point of the Hartley transform represents the average value of the transformed spectrum, and in the case of EEM always has a positive value.

For the rest of the points in the transformation if a particular point has been encoded into a 1, the next point is encoded into a 1 only if the two points are comparable in value. A point is encoded into a 0 if it is significantly smaller than its predecessor. On the other hand, if a point has been encoded into a 0, the next one is also encoded into a 0, unless it is significantly larger, in which case it is encoded into a 1. Significantly smaller, or larger, is defined as being at least 25% smaller, or larger, respectively. An example of clipping method A can be seen in Figure 23.

For the second relative encoding method tested, clipping method B, three symbols are utilized for the encoding, 1, 0, and -1. The value of each point is compared against the value of its two neighboring points, the one immediately proceeding, and the one immediately following.

If a particular point of the transformation is a local maximum, i.e. the value of the point is larger than both neighboring points, it is clipped into a 1. If it is a local minimum, i.e. smaller that both neighboring points, it is clipped into a -1. The remaining points are clipped into

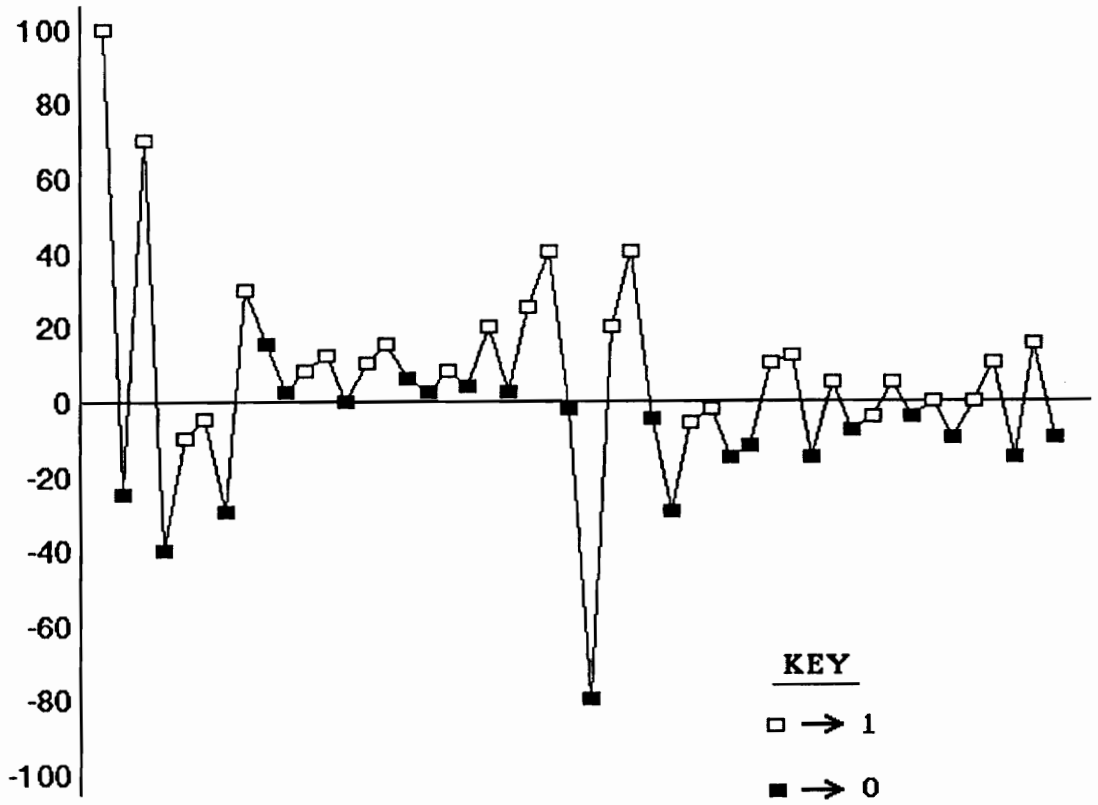


Figure 23. Example of clipping method A.

0's. Figure 24 shows an example of clipping method B.

The third and last relative encoding method developed and tested, clipping method C, also utilizes three symbols, 1, 0, and -1. Similar to method B, a particular point is clipped into a 1, or a -1, if it is a local maximum or minimum, respectively.

In this method, however, the remaining points of the transformation are clipped into 1's if they are significantly close to a local maximum, or they are clipped into 0's if they are significantly close to a minimum. Points which are not significantly close to a local maximum nor a minimum are clipped into 0's. Significantly close is defined as being equal or larger than 75% of the closest maximum; or being equal or smaller than 75% of the closest minimum, respectively. Figure 25 shows an example of this method.

The final effort in optimizing the results of the reverse search of the library was an attempt to eliminate problems derived from the presence of white noise in the low spatial frequencies of the Hartley transform.

As was demonstrated in the Hartley transform theory section, white noise present in the original spectrum will be significantly reduced in the transform domain, but yet it will continue to be spread over the entire transform domain,

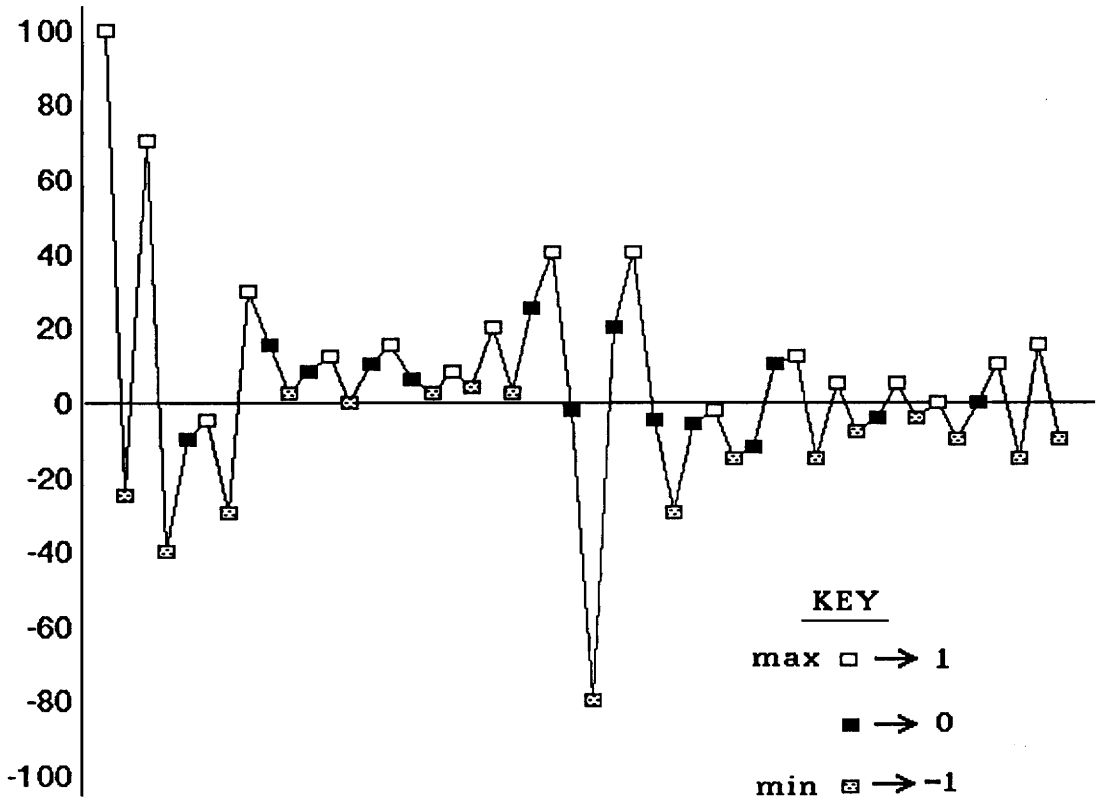


Figure 24. Example of clipping method B.

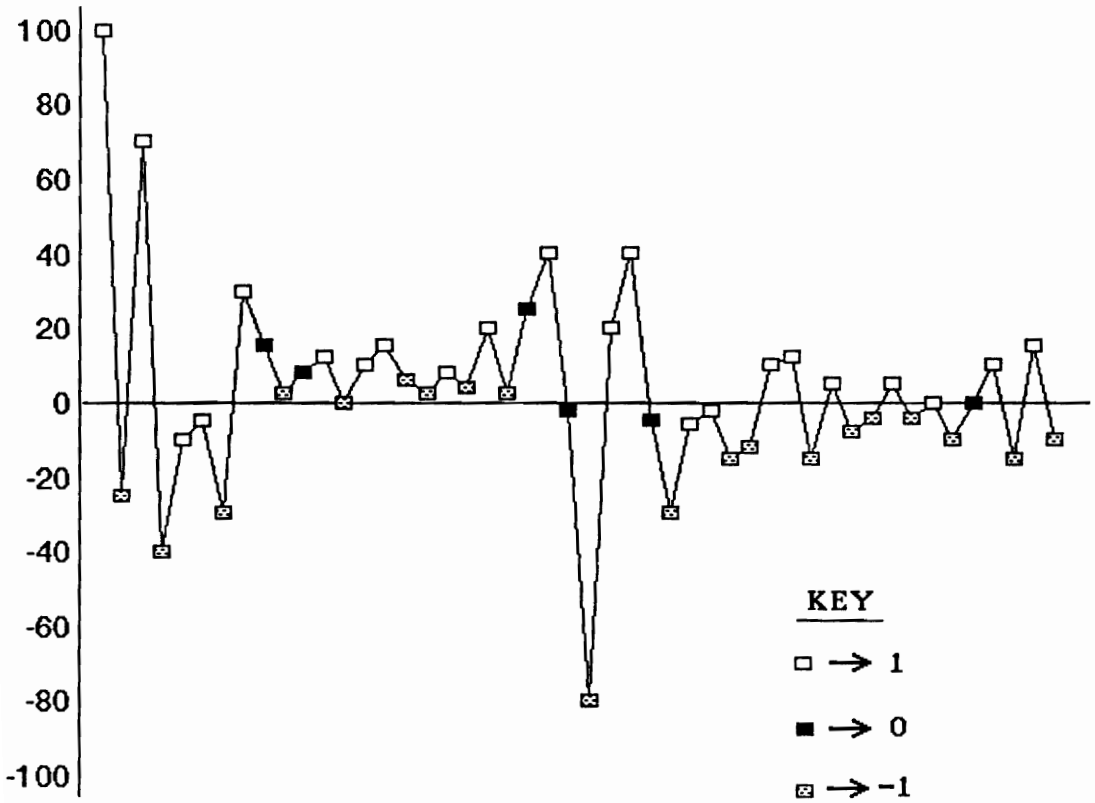


Figure 25. Example of clipping method C.

even among the low spatial frequencies. The presence of that noise in the HTL files could potentially cause problems during the spectral encoding process.

The magnitude of several points in the HTL files could be changed so that during the encoding step the clipped pattern of the entire spectrum could be significantly altered, to the point where the identification of the compound would not be possible. Points most vulnerable to this effect would be low magnitude points. Especially in the case of the zero-crossing algorithm, the value of points close to zero could be forced to change from positive to negative, and vice versa, thus changing the clipped pattern of the spectrum.

To overcome this problem, the higher spatial frequencies of the Hartley transform were used to estimate the noise level in the transform domain. Furthermore, low spatial frequency points with magnitudes equal or smaller to the estimated noise level were forced to zero. By doing so, those points could not further alter the clipped pattern of the spectrum, and thus effect the results of the library search.

As was explained previously, the effort of this part of the study was to find the optimum unfolding and spectral encoding methods to be employed with the algorithm. To do so, the unfolding methods (boustrophedon and spiral) and the

encoding methods (zero-crossing and the three relative encoding methods A, B, and C) explained above, as well as combinations of the two, had to be tested in order to selected the optimum approach.

All possible combinations of those methods, as well as the attempt to filter the Hartley transform, can be seen in Figure 26. In Figure 26, the two dimensional Hartley transformation was also included and tested. The main reason for including the two dimensional transformation in this part of the study was to check the advantage of the unfolding and the application of the one dimension Hartley transformation over the two dimensional transformation.

Also in Figure 26, the number of points saved at each step of the process can be seen. Again, in the case of the encoding methods, the number in parenthesis denotes the length of the clipped patterns that would emerge from the encoding. It should be noted that in the case of the two dimensional transformation the number of points stored is higher than for the other combinations, 576 points instead of 512. This is a result of the way the spatial frequencies are represented in the two dimensional transformation.

Now that all those combinations have been formed, in order to find the optimum one, a reference library for each of them has to be developed. Then, the algorithm would be employed to do the reverse search of each of those libraries

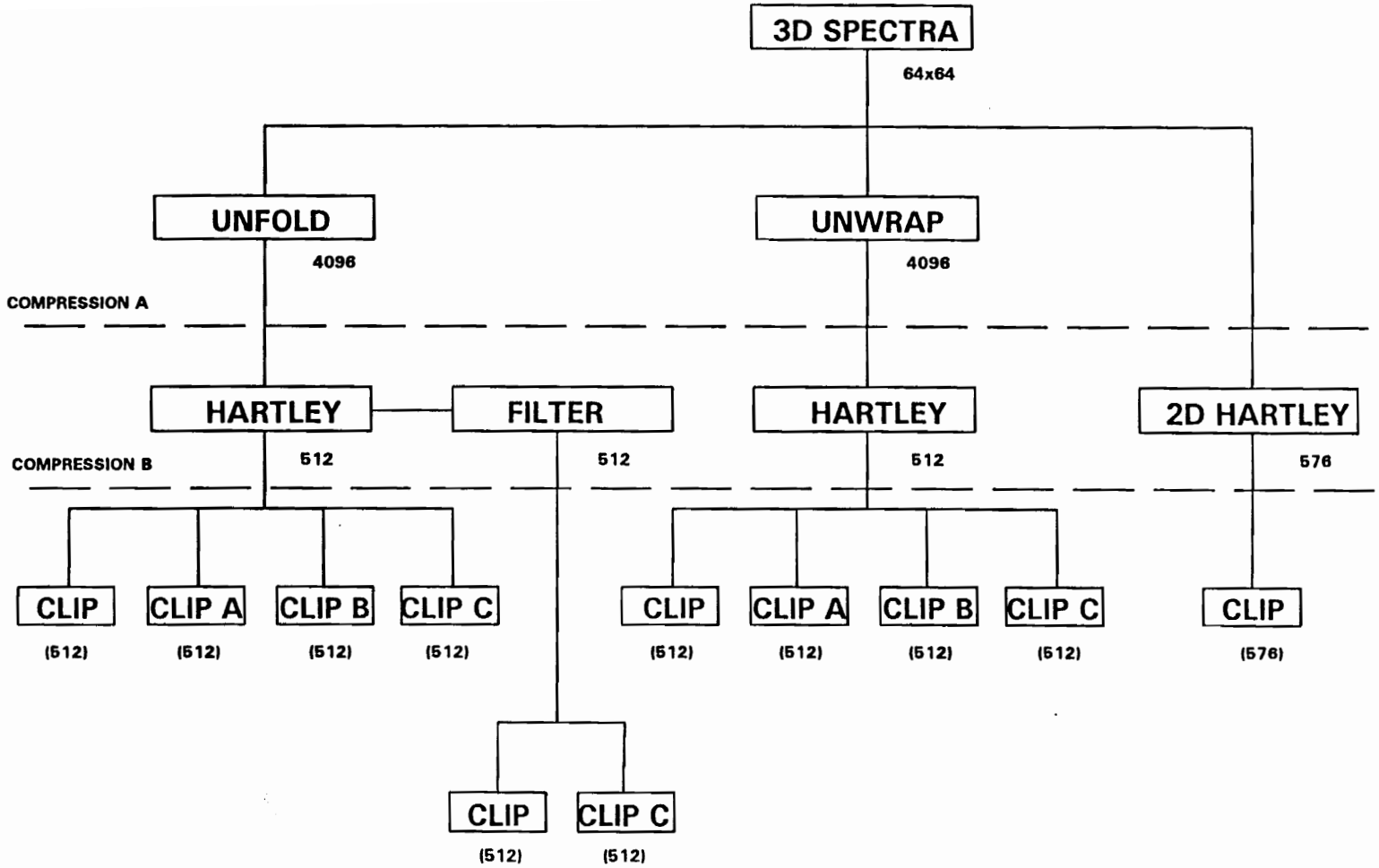


Figure 26. Combinations tested to find the optimum unfolding and clipping procedures.

for a specific set of unknown mixtures. The combination which would yield the best result during the reverse search, would be the optimum one.

For this purpose, the Excitation Emission Matrices of 61 poly-aromatic hydrocarbons, collected during a previous study,[60] were selected to develop the different reference libraries. A list of those compounds, along with their code names can be seen in Table III. The use of two letter code names for referring to the library members was adopted from the data collection software used in running the fluorescence spectrophotometer.

Several of those reference spectra were randomly selected to form the set of unknown mixtures. In order to form a representative set of possible unknowns that the algorithm would encounter in a real life application, and also to be able to draw statistical conclusions from the results, a large number of unknown mixtures had to be tested. Also the unknown mixtures should have varying number of components with different relative concentrations.

To achieve that goal, 40 different unknowns were made by randomly selecting members of the reference library. Ten of these unknowns were constructed with only one component (10 unknown x 1 component = 10 components), ten others with two components each (10 unknown x 2 components = 20

Table III. List of compounds in the reference library.

AA	2-Aminoanthracene	MN	2-Methylnaphthalene
AC	Anthranilicacid	NA	Naphthalene
AN	Anthracene	ND	aNPD
AQ	Anthraquinone	NO	2-Naphthol
AR	Acridine	NP	aNPO
AZ	Azulene	PA	9-Phenylanthracene
BB	BBOT	PB	PBD
BD	BBD	PD	PPD
BE	4-Biphenylphenylether	PE	Perylene
BI	2,2-Binaphthyl	PH	Phenanthrene
BN	bNPD	PN	2-Phenylnaphthalene
BO	BBO	PO	POPOP
CA	9,10-Dichloroanthracene	PP	PPO
CH	Chrysene	PQ	Phenanthrenequinone
DA	9,10-Diphenylanthracene	PY	Pyrene
DE	1,1-Diphenylethylene	QP	p-Quaterphenyl
DI	4,5-Diphenylimidazule	QU	Quinoline
DN	2,3-Dimethylnaphthalene	SA	Salicylic Acid
DP	DimethylPOPOP	SN	2,6-Dimethylnaphthalene
DS	Diphenylstilbene	TA	Triphenylamine
EA	Methylanthracene	TE	Tetracene
ES	Esculin	TM	Triphenylamine(1.00e-3)
FL	Fluorene	TN	Triphenylamine(5.00e-4)
IA	1-Aminoanthracene	TP	1368-Tetraphenylpyrene
IB	1,1-Binaphthyl	TQ	Triphenylamine(1.00e-5)
ID	Indole	TR	Triphenylene
IM	1-Methylnaphthalene	TS	Triphenylamine(1.00e-4)
IN	1-Naphthol	TT	Triphenylamine(5.00e-5)
IP	1-Phenylnaphthalene	VA	9-Vinylanthracene
MA	9-Methylanthracene	VB	4-Vinylbiphenyl
MB	4-Methylbiphenyl		

components), ten more with three components each (10 unknown x 3 components = 30 components), and finally, the last ten were constructed with five components each (10 unknown x 5 components = 50 components), giving a total number of 110 components.

The first ten unknowns, with only one component, will be referred to as Unknowns 1 through 10, where the remaining unknowns, with more than one components, will be referred to as Mixtures 1 through 30.

Each reference spectrum, before being tested as an unknown, was also multiplied by a random concentration factor. The composition of those 40 unknowns, along with the random concentration factors each component was multiplied by may be found in Appendix B.

In order to study the effect that noise would have on the algorithm in the case of one component unknowns random noise was added to each of these spectra. The peak to peak value of the added noise was 0.05 fluorescence intensity units. For the worst case (Unknown 10) the level of the noise added gave a signal to noise ratio, S/N, of 20:1. The EEM of Unknown 10, as well as the EEMs of the rest of the unknown mixtures, can be seen in Appendix C.

At this point, each of those 40 unknowns were unfolded, transformed, and clipped, according to each of the unfolding-clipping combinations explained above. The

library CLP files were then searched against the generated CLP files of the unknowns.

The outcome of a library search for an unknown mixture, as explained in the section describing the algorithm, is a Positive/Negative Ratio for each member of the spectral library. The higher the Ratio, the higher the probability of the presence of the reference in the unknown mixture.

An example of the library search results is shown in Table IV. The table shows the results for the Mixture 3 unknown. In the first column of the table the code names for the 61 library members are shown. In the next three columns the number of Positive, Non-applicable (N/A), and Negative Points, respectively appear. In the last column are the calculated Positive/negative Ratios for each library member. At the bottom of the table, it can be seen that the spectral encoding method tested in this particular example was clipping method C.

A closer examination of the last column reveals the expected result that the great majority of the reference spectra have a ratio of one, or very close to one. Only very few have ratios significantly higher than one. This observation becomes even more obvious by examining a plot of those Positive/Negative Ratios, along the y-axis, versus the 61 reference spectra, along the x-axis, as shown in Figure 27.

Table IV. Library search results for unknown Mixture 3.

SPECTRUM	POSITIVE POINTS	NA	NEGATIVE POINTS	POSITIVE/NEGATIVE RATIO
aa	156	75	171	0.9123
ac	161	100	202	0.7970
an	148	95	196	0.7551
aq	217	100	142	1.5282
ar	265	55	69	3.8406
az	190	103	176	0.0795
bb	148	81	172	0.8605
bd	169	108	211	0.8009
be	146	65	145	1.0069
bi	183	104	145	1.2621
bn	184	106	166	1.1084
bo	174	93	165	1.0545
ca	147	90	156	0.9423
ch	163	98	188	0.8670
da	128	73	149	0.8591
de	141	88	153	0.9216
di	210	98	122	1.7213
dn	165	85	156	1.0577
dp	161	79	137	1.1752
ds	189	91	153	1.2353
ea	161	99	158	1.0190
es	188	100	169	1.1124
fl	142	72	132	1.0758
ia	141	81	153	0.9216
ib	189	101	162	1.1667
id	142	71	131	1.0840
im	179	100	151	1.1854
in	176	95	169	1.0414
ip	149	101	153	0.9739
ma	201	64	107	1.8785
mb	140	76	120	1.1667
mn	171	91	146	1.1712
na	167	86	154	1.0844
nd	172	102	209	0.8230
no	157	106	188	0.8351
np	170	108	186	0.9140
pa	214	79	112	1.9107
pb	175	106	170	1.0294
pd	150	81	147	1.0204
pe	169	78	156	1.0833
ph	166	93	152	1.0921
pn	176	100	172	1.0233
po	149	101	188	0.7926
pp	170	106	192	0.8854
pq	198	86	130	1.5231
py	300	81	66	4.5455
qp	136	83	118	1.1525
qu	190	98	150	1.2667
sa	204	99	151	1.3510
sn	167	93	144	1.1597
ta	169	97	134	1.2612
te	135	83	138	0.9783
tm	188	104	183	1.0273
tn	186	108	168	1.1071
tp	131	70	112	1.1696
tq	147	70	134	1.0970
tr	180	102	175	1.0286
ts	175	97	181	0.9669
tt	168	101	197	0.8528
va	128	77	165	0.7758
vb	157	77	148	1.0608

Search limits : 1 to 512 SEARCH C

Each asterisk in Figure 27 represents the Positive/Negative Ratio for one reference spectrum. The two points on the graph shown as solid squares represent the Positive/Negative Ratio for the two reference spectra which were the actual components of the mixture. Clearly, the plot shows that the reverse search was successful.

On examining a second example, though, the results are not quite so clear. Figure 28 shows the results of the reverse search of the library for a three component mixture, Mixture 19. Using the same notation as before, solid squares for the actual components of the mixture and asterisks for the other members of the library, the plot reveals that only two of the three components could be picked out during the search. They have large Positive/Negative Ratios. The third component would be lost, with a Ratio very close to one.

A second important observation can be made from these two plots of the above examples. Besides the actual components in the unknown mixtures, several other members of the library, were found during the search to have Ratios significantly higher than one.

A conclusion can be drawn from the above observations. The comparison metric used in this reverse search of the library is the Positive/Negative Ratio which is a continuous

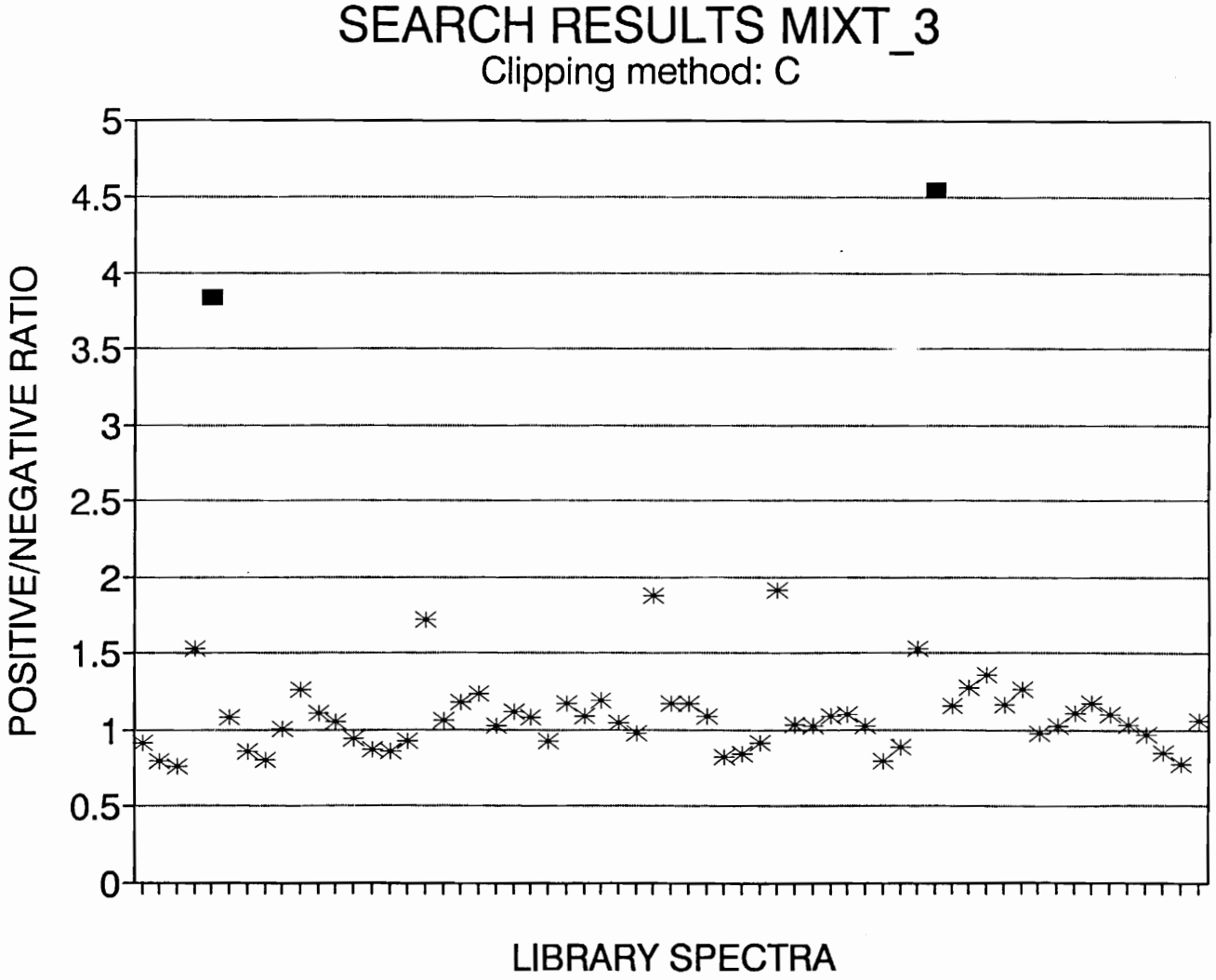


Figure 27. Search results for Mixture 3 (two components).

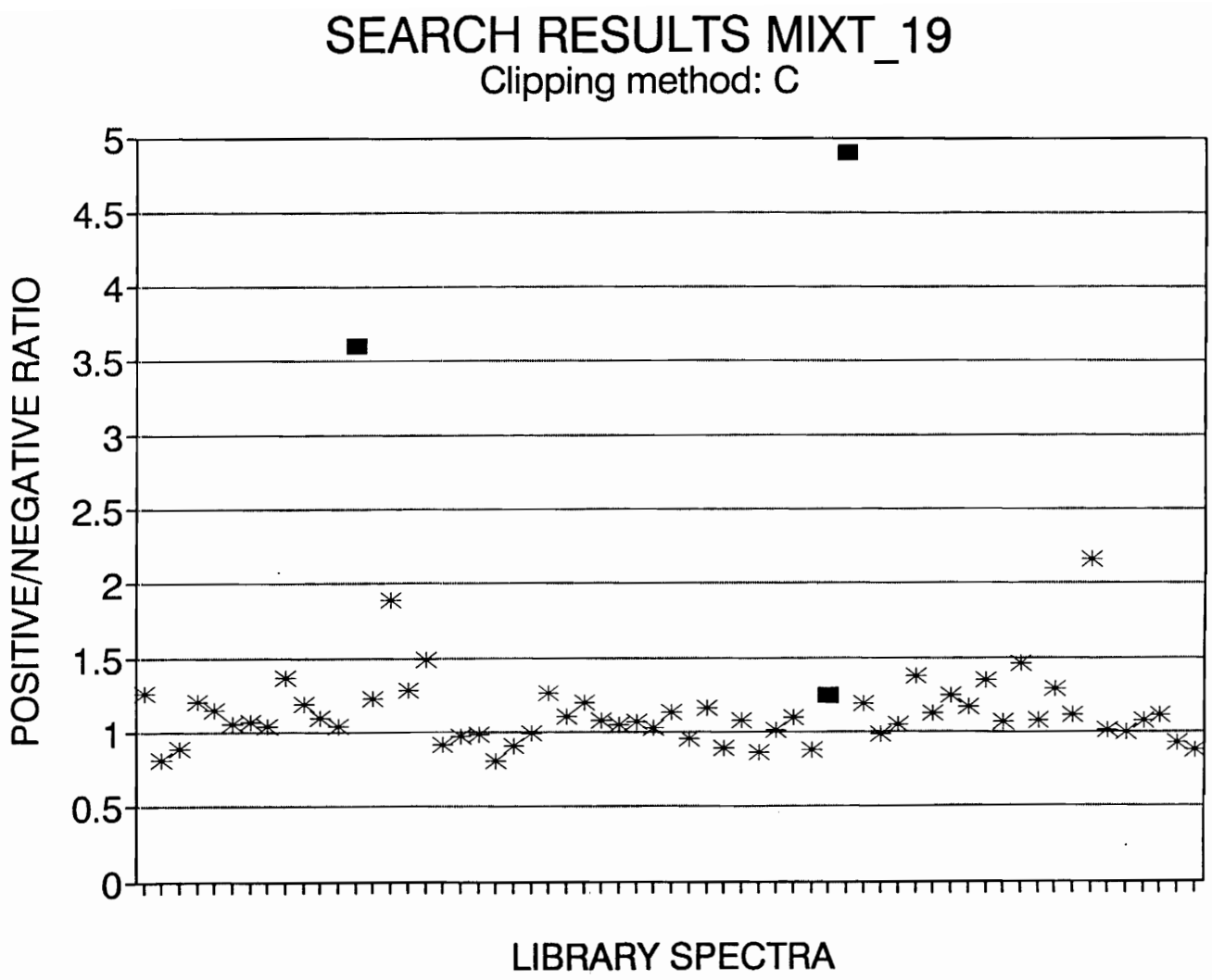


Figure 28. Search results for Mixture 19 (three components)

type variable. A threshold value needs to be established. Reference spectra with Positive/Negative Ratio values above this threshold would pass the filtering step and would be considered possible components of the unknown mixture. Reference spectra with Ratios below the threshold would be filtered out and would not be considered during the next step of the algorithm, the NNLS method.

In order to find this threshold value, the means and the standard deviation for the reference spectra for each of the 40 unknowns were calculated. Figure 29, and Figure 30 show the calculated mean and standard deviation for each unknown.

As expected, the mean of the Positive/Negative Ratios of the reference spectra for each one of the 40 unknowns was very close to one. The actual average value for the means shown in Figure 29, was about 1.1. The corresponding standard deviations, shown in Figure 30 were relative small with an average value of about 0.2.

An appropriate threshold value for the Positive/ Negative Ratio applied during this type of work was 1.5, which is approximately the mean plus two standard deviations. Statistically, this is a 95% cut off value, i.e. only about 5% of the library members are expected to pass the filtering step and will be selected as possible components of an unknown mixture after the library search.

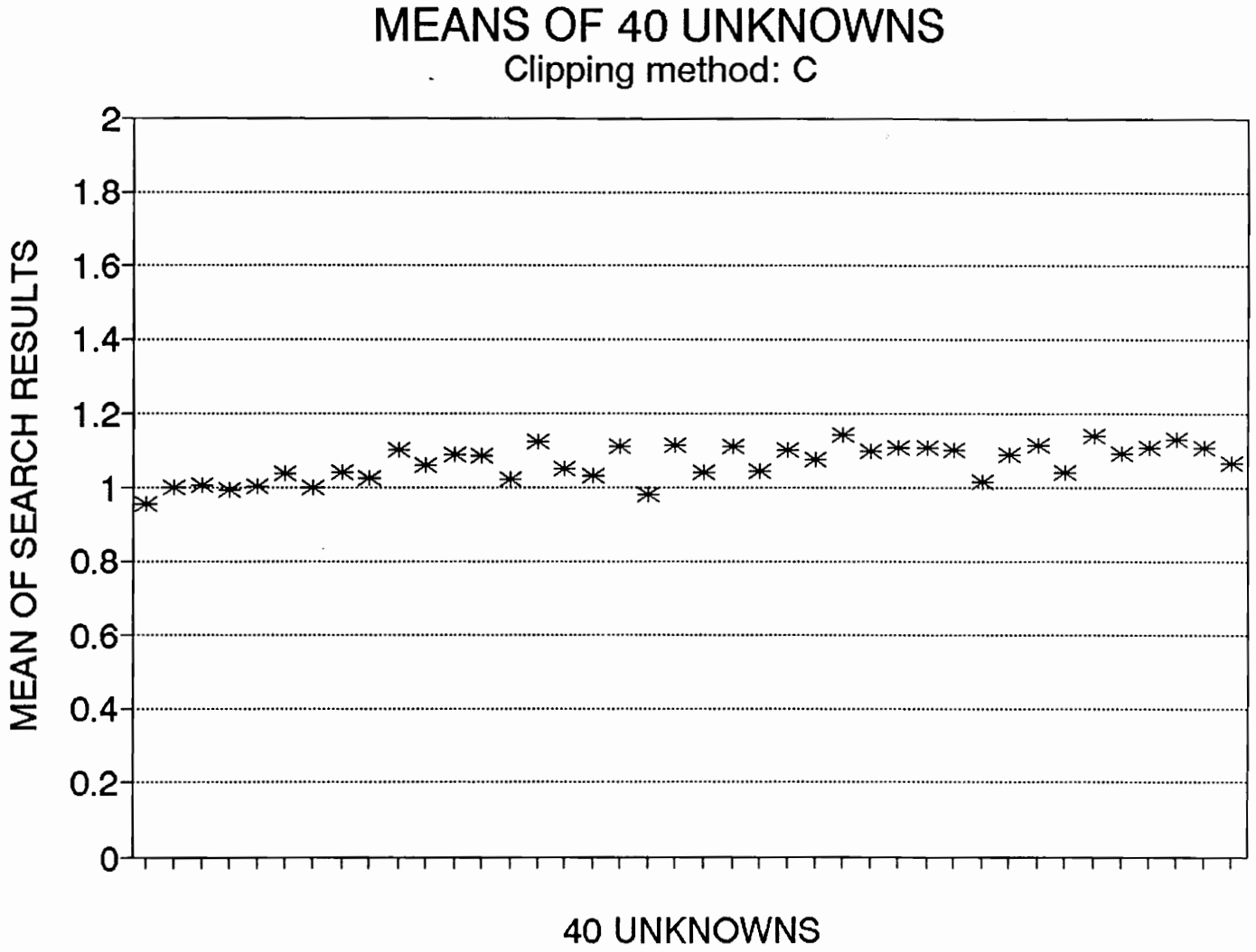


Figure 29. Means of the 40 unknowns - Clipping method C.

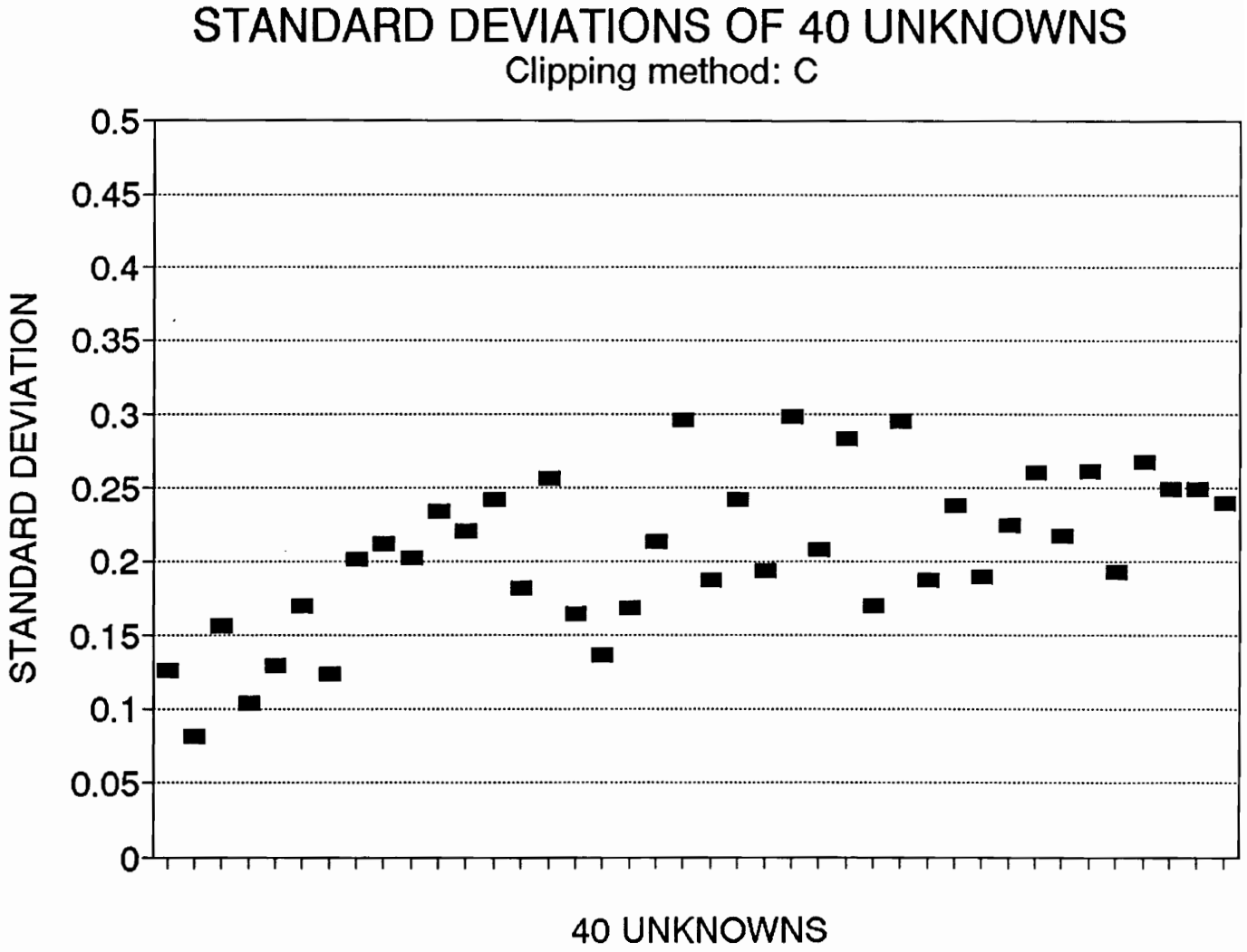


Figure 30. Standard Deviations of the 40 unknowns - Clipping Method C.

Because of the relative small size of the spectral library employed in this study, a 95% cut off point would only allow a very small number of spectra, (about 3 reference spectra) to pass this step of the algorithm. This number is obviously smaller than the maximum number of components (five), in the mixtures that the algorithm will be tested against.

For that reason, during the reverse library search the six spectra with the highest ratios will always be selected as possible components of the unknown mixture, regardless of their actual Ratio.

For the rare, but always possible situation where a very large number of reference spectra would have Positive/Negative Ratios higher than 1.5, the number of compounds selected as possible components will be limited to 10. This corresponds to about 15% of the total number of reference compounds in the library.

To summarize the rules which will be used to select the reference compounds to be considered as possible components of the unknown mixtures :

1) Reference compounds with a Positive/Negative Ratio of over 1.5, or the six compounds with the highest ratios, will be selected.
2) If more than 10 reference compounds have ratios of over 1.5, only the first 10 will be

selected.

It should be noted that although the above examples and calculations for the estimation of a threshold value, and the generation of the selection rules, were developed using only one of the unfolding-clipping combinations tested, (Figure 26) similar observations were found for the remaining approaches. Consequently, the same threshold value and selection rules were used for the reverse search of the reference libraries developed for all the combinations.

The present effort continues to be directed towards the selection of the optimum unfolding-clipping combination.

At this point (1) the reference libraries have been developed, (2) the 40 unknowns have been searched against these libraries, and (3) the selection rules have been established. Now a list of the reference spectra that were selected for each one of the tested combinations can be developed. Table V shows an example of such a list, for the boustrophedon unfolding - clipping method C combination (the composition of the 40 unknown may be found in Appendix B).

The Table shows for each one of the 40 unknowns which reference compounds were selected by the algorithm as possible components after the reverse search. The selected compounds for each unknown are ranked from highest to lowest Ratio. Also shown in the same Table are the code names of

Table V. Example of a list with selected reference spectra.

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
UNKNOWN										
1	**an**	bo	ea	po	bn	pb				
2	**bb**	da	va	dn	tp	de				
3	**bd**	az	nd	aq	ia	ph				
4	**fl**	mb	pd	vb	tq	dn				
5	**ma**	pa	ar	dp	da	pb				
6	**mn**	dn	im	pd	ip	na	vb			
7	**pd**	vb	na	dn	mn	qu				
8	**po**	bo	ds	ea	an	ac				
9	**pp**	tm	aq	bn	tn	tr				
10	**tq**	tt	ts	qp	ch	qu				
MIXTURE										
1	**in**	sn	**bb**	pq	da	va	ib			
2	**py**	pq	sa	di	ta	aq				
3	**py**	**ar**	pa	ma	di	aq	pq			
4	**ib**	**an**	qp	in	pq	ip				
5	**in**	sn	**tm**	pq	aq	tn	pp	tr	bn	no
6	**az**	bd	**va**	bb	da	tm				
7	**ca**	**id**	tp	be	da	de				
8	**bb**	da	va	**aq**	tm	py				
9	**np**	**qp**	pn	ds	ib	di				
10	**qu**	**qp**	ib	di	tt	tq	ip			
MIXTURE										
11	**da**	bb	va	tp	ca	**pn**				
12	**pd**	vb	na	**ib**	**te**	sn	qu	dn		
13	**fl**	**bo**	po	an	ds	ea				
14	**tn**	tr	no	ts	pb	bn	sa	bi	aq	**pa**
15	**im**	ip	**pe**	nm	im	te				
16	**qu**	**pb**	no	sa	tn	ts	tr	bn	tt	py
17	**ph**	**ca**	tp	da	ta	**be**				
18	**bi**	pb	sa	tn	**vb**	tr	no	bn	pb	ts
19	**ph**	**ca**	tp	da	di	ta				
20	**ts**	tt	tn	no	tr	ch	tq	pb	bn	tm
MIXTURE										
21	**pp**	tm	**qp**	aq	ib	**an**				
22	**pe**	**pd**	vb	na	**bd**	**mn**	az	dn	**te**	
23	**tt**	**in**	ts	no	tq	ch	tn	im	ip	sn
24	**an**	**ea**	bo	**aa**	be	**id**				
25	**dp**	**ts**	tt	tr	tn	no	qu	bn	ma	
26	**pd**	vb	na	**bd**	ib	mn				
27	**sa**	pb	**bi**	pq	tr	py	ta	bn	tn	**aq**
28	**in**	**py**	pq	sn	sa	tn	bn	aq	**tr**	pb
29	**ib**	**tm**	aq	pp	tn	**az**	ip	**aa**	in	tr
30	**tr**	tn	ts	bn	pb	no	tm	sa	aq	**ch**

Number of correct compounds retrieved : **80**
Total number of compounds retrieved : 292

the actual components in the unknown mixtures, selected during the search. These appear as underlined and highlighted entrees in the table.

Examining the table more carefully, it is evident that for all the unknowns for which the library was searched the reference spectrum with the highest Ratio was always an actual component of the unknown. Furthermore, for the unknowns with more than one component, in most situations, the reference spectrum with the second highest Ratio was also an actual component of the unknown.

The same two examples used to explain the library search, Mixtures 3 and 19, will be used again to further clarify the entries in Table V. First, in the case of Mixture 3, which is a two component unknown mixture, it can be seen that according to the selection rules seven reference compounds were selected as possible components:

Mixture 3 **py** **ar** pa ma di aq pq

The compound shown on the left, py, had the highest Ratio, and the one on the right, pq, had the lowest Ratio.

It can be seen that the two actual components of the Mixture 3, reference spectra py and ar, were found to have the highest Positive/Negative Ratios among all the reference spectra. This, of course, is the same observation made by examining the plot of the library search results in Figure 27.

In the case of the second example, Mixture 19, which is a three component mixture, only six reference spectra were selected :

Mixture 19 **ph** **ca** tp da di ta

Six is the minimum number of reference spectra that could be selected during the library search, according to the selection rules. Again, the compound on the left, ph, had the highest Ratio, and the one on the right, ta, the lowest Ratio. In this example, although the two components with the highest Ratios were also actual components of the mixture, the third component was not retrieved from the library.

In an effort to select the optimum combination of unfolding and spectral encoding procedures (Figure 26) for the final algorithm, the above examples will be further explored.

The number of actual components in the unknowns that were successfully selected and retrieved during the library search, which equals the number of highlighted and underlined spectra in Table V, is obviously an important criterion of the performance of a particular combination of unfolding-clipping techniques. Since the total number of components in the 40 unknowns, as explained previously, was 110, the closer the number of selected reference spectra is to 110, the better the performance.

A second criterion of the performance of the reverse search is the total number of reference spectra selected during the search. Since according to the selection rules at least six reference spectra would be retrieved for every unknown regardless of their Positive/Negative Ratio, the minimum number of reference spectra that can be selected for all 40 unknowns, is 240 (6 reference x 40 unknown). Thus, the closer the total number of selected reference spectra to 240, the better the performance of the library search.

Both of these criteria are equally important for a comparison of the performances of the different combinations. They represent recovery rate and specificity of the search. For that reason, the Success Index which scores the performance of the combinations employed, is defined as the ratio of the two :

$$\text{Success Index} = \frac{\text{Number of actual components retrieved}}{\text{Total number of components retrieved}}$$

From the definition, it can be seen that the lower limit of the Success Index is zero. This would occur when no components were correctly retrieved. The upper limit, can be found when the number of correctly retrieved compounds equals the total number of components (110), and the total number of compounds retrieved equals the minimum possible number of retrieved compounds (240). This upper limit would be 0.458 (=110/240).

To avoid this strange upper limit, the Success Index was normalized, by dividing it with its maximum value, (0.458) :

$$NormalizedSuccessIndex = \frac{SuccessIndex}{0.458}$$

The lower and upper limits for the Normalized Success Index, which will rate the performance of the different combinations of unfolding and spectral encoding are 0.0 and 1.0, respectively.

For the combination shown in Table V, boustraphedon unfolding and relative clipping method C, the number of correctly retrieved components (number of highlighted and underlined spectra) was 80. The total number of compounds selected from the library was 292. Thus the Normalized Success Index is 0.60.

At this point, some further explainations on the purpose and use of the Success Index are in order.

The purpose of the Success Index is to establish a common ground on which the different library development schemes presented previously can be compared. The Index is intended to reflect how the different schemes perform, i.e. how successful the different library development schemes are on retrieving the appropriate compounds. The name of the Index was derived from this function.

The Index should only be used as a relative measure of performance for the different library schemes, and should not be used as a percentage measure of the performance of the complete algorithm. The performance of the complete algorithm will be evaluated later.

The Normalized Success Index for all the combinations explained previously (Figure 26) was calculated. The results are shown in Figure 31. The calculation of the Normalized Success Index for the different library development schemes is based on a single set of randomly selected unknown mixtures, thus the calculation of a Standard Error is not possible. On the other hand, the large number of unknown mixtures as well as the random selection of the set of the unknown mixtures, allows us to use the single measure of the Success Index in the comparison of the different library development schemes. Similar approaches have been used in the past by other researchers in the area.[77]

Since the Success Index depends on two independent - variables, in order to better judge the performance of the various library development combinations, Figure 31 also shows the number of actual components retrieved from the library search on a percentage basis.

For example, a 73% figure of merit means that during the first library search performed by the algorithm, 73% of

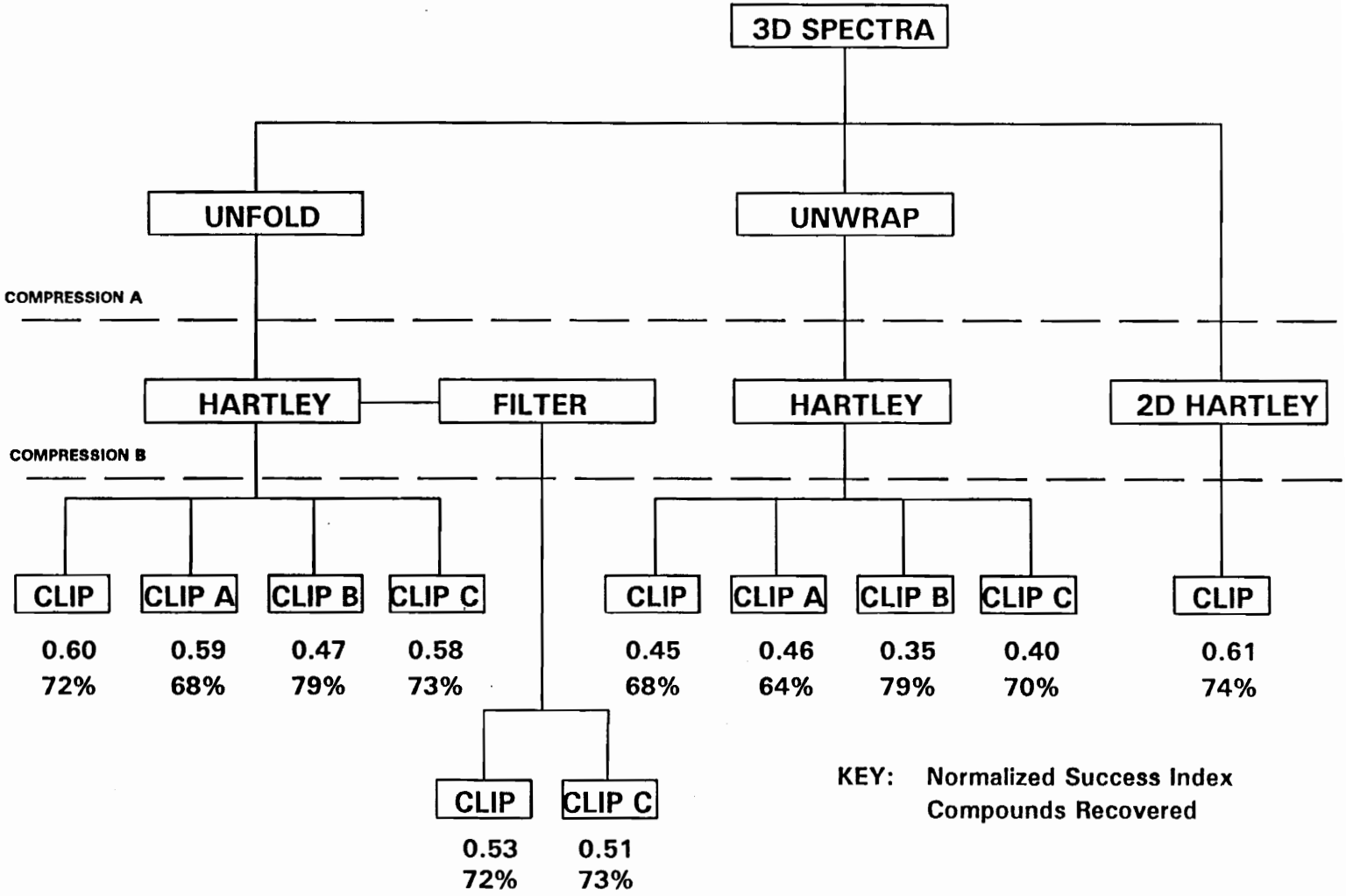


Figure 31. Normalized Success Index of different library development schemes.

the components in a mixture would be expected to be retrieved. The remaining will be retrieved during following iterations through the loop of the algorithm.

To help in a comparison of the different library development schemes tested in the presented study, Figure 32 shows a bar graph of the calculated Normalized Success Indexes. Several conclusions on the performance of the various methods for the unfolding of three dimensional spectra and spectral encoding can be drawn from that graph.

First, is the obvious advantage of the boustrophedon unfolding (crossed lines pattern) over the spiral unwrapping (dotted pattern). From the value of the Normalized Success Index it can be seen that for every clipping method, the boustrophedon unfolding always out-performed the spiral unwrapping. In every case, the combinations involving boustraphedon unfolding gave higher Normalized Success Indexes than those involving the spiral unwrapping.

Apparently, in spiral unwrapping the linear array which is created is not as symmetrical as that resulting from boustrophedon unfolding. This asymmetry forces the transformation to spread the signal over more spatial points, so the 512 points which are saved are not enough to distinguish as succesfully members of the library during the library search process.

Comparing the performance of the two clipping methods

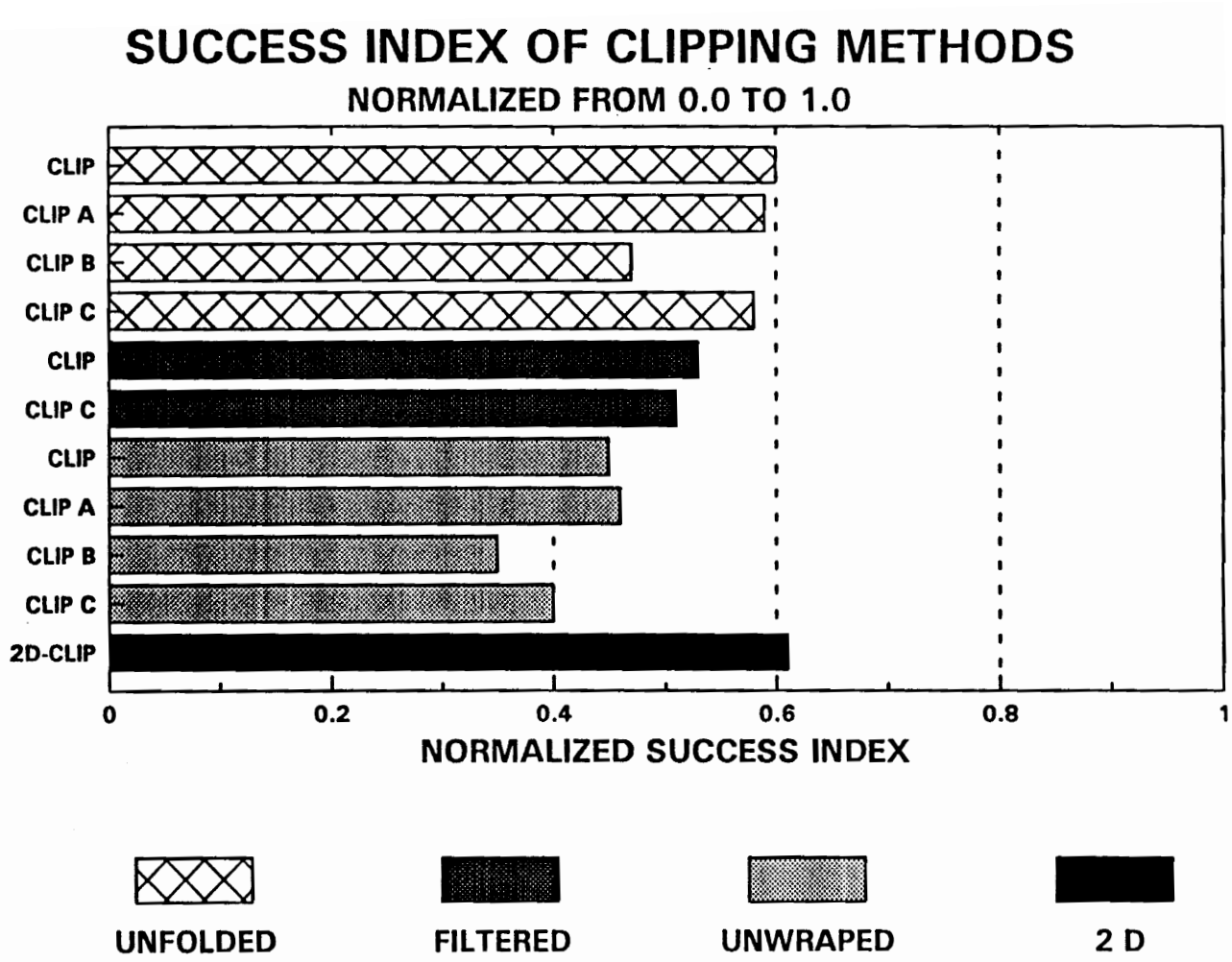


Figure 32. Bar graph of the Normalized Success Indexes.

[zero-crossing and clipping method C, with filtering (dark gray pattern) and without filtering (crossed lines pattern) of the Hartley transform], it is apparent that filtering hinders the performance of the clipping methods. The observation implies that there is significant information about the characteristics of the reference spectra in all the low spatial frequency points of the Hartley transform, even those with very small magnitude.

Comparing the different spectral encoding method one observes an obvious disadvantage of the relative clipping method B. Every combination involving clipping method B gave small Success Index. The performances of the remaining three methods, especially in the case of unfolded spectra, were similar enough that all appear to have equal ability.

Although the zero-crossing algorithm is the simplest one, it compares favorably with other relative encoding methods which reflect finer structures in the transformed spectra. This satisfactory operation of the zero-crossing clipping algorithm results from the fact that the data in the transform domain is mean centered at zero.[70] The observation that the simplest clipping algorithm gives better or equal results to more complicated methods suggests that the Occam's Razor be applied.

Comparing the one dimensional Hartley transform with

the two dimensional transform, Figure 32, it appears that the two dimensional Hartley transformation combined with the simple zero-crossing spectral encoding method out-performs the one dimensional transformation. Such a conclusion would be false. First, recall that in the case of the two dimensional transform the number of points saved after the Hartley transformation (576 points) was larger than the number of points saved for the rest of the combinations (512 points). Second, the Normalized Success Index is not the only critirion on the performance of these combinations. The advantages of the unfolding of the spectra, which were explained in previous sections, are also very important.

Finally, to select the optimum combination, the bar-graph of the Normalized Success Indexes (Figure 32), as well as the percentage of the correct number of components retrieved (Figure 31) in each case can be examined concurrently.

It can be seen that boustrophedon unfolding combined with relative clipping method C gives one of the highest-Normalized Success Indexes as well as one of the highest percentages of correctly retrieved components. That combination will be utilized in the next part of the study, which involves the application of the NNLS method. During that part the identification of the components of the unknowns will be verified, and their concentrations will be

estimated. The qualitative and quantitative analysis of the unknowns will then be complete.

2. APPLICATION OF THE NNLS METHOD

Since boustrophedon unfolding combined with clipping method C gave the best results during the library search part of the algorithm, the library developed for that combination will be used during the next step, the NNLS method. The results of the reverse search of that library are shown in Table V.

For this part of the algorithm, the HTL files are employed. Specifically, the HTL library files of the reference spectra which were retrieved during the library search, along with the HTL file of the unknown mixture, will be used in the NNLS calculations.

The expected result of the method is the estimation of the concentration factors of the components of the mixture. For the reference compounds which were retrieved during the library search, but which were not actual components of the unknown mixture, the concentration factors are expected to be equal to zero.

Furthermore, the absolute concentrations of the components can be computed from those factors. The absolute concentration of each component, would be equal to the concentration of the solution of the pure compound used to

collect the reference spectrum, times the corresponding concentration factor.

The relative accuracy of the concentrations of the solutions of the reference compounds involved only two significant figures. This also limits the accuracy of the calculated concentration factors. For that reason, and also in order to decrease the time required to perform the computations (the number of computations in the NNLS method increases exponentially with the number of points involved), only a small number of the points in the HTL files were entered in the calculations. In fact, only the first 128 points of the Hartley transform were involved. The number 128 was small enough to keep the time required to perform the computations to a minimum, and at the same time large enough to give the required accuracy.

To serve as examples of the type of results obtained during this step, Mixtures 3 and 19 will again be explained in details.

The HTL files of the seven reference compounds - retrieved from the library, namely py, ar, pa, ma, di, aq, and pq, along with the HTL file of Mixture 3, were entered in the NNLS calculations. As explained earlier, a FORTRAN implementation of the method, was used. The output of the NNLS method for Mixture 3, is shown in Table VI.

The first column of the output shows the two letter

Table VI. Output of NNLS calculations for Mixture 3.

NON-NEGATIVE LEAST-SQUARES RESULTS FOR : MIXTURE 3

REFERENCE SPECTRUM	CONCENTRATION FACTOR	(ACTUAL)	
1) aq	0.00	(0.0)	
2) ar	2.00	(2.0)	*
3) di	0.00	(0.0)	
4) ma	0.00	(0.0)	
5) pa	0.00	(0.0)	
6) pq	0.00	(0.0)	
7) py	1.00	(1.0)	*

RNORM = 8. MODE = 1

****

NOTE : The asterisk denote the actual components of the unknown.

code names for the reference spectra entered in the method, and the second column shows the calculated concentration factors. The numbers in parenthesis next to the concentration factors, are the actual factors which were used to develop the unknowns.

At the bottom of Table VI the value of the RNORM for the estimated solution can be seen. As previously explained, the value of RNORM serves as an indication of the quality of the calculated solution : small RNORM values indicate an accurate and acceptable solution.

Since the RNORM for Mixture 3 had a very small value, eight, the estimated concentration factors should be very accurate. This becomes obvious by comparing the calculated factors with the actual factors shown in parenthesis.

The last item shown at the lower right side of Table VI is the mode at which the execution of the program carrying the NNLS calculations terminated. Mode equal to one indicates normal termination of the program. Any mode value different from one indicates the detection of an error. - Errors can be generated either during the initiation of the program, e.g. fewer data points are found in the input file than expected, or during the execution of the program, e.g. attempt to divide by zero.

As anticipated for Mixture 3, since all of the components were retrieved during the library search, the

NNLS method had no problem to correctly and very accurately estimate the concentration factors of the components of the mixture. The analysis was completed during the first NNLS calculation, thus the algorithm never entered into the loop.

In the case of the second example, Mixture 19, the situation is different. As can be seen in Table VII, the RNORM had a very large value, indicating that the calculated solution is not an acceptable one. This outcome is obviously expected since during the library search only two out of three components were retrieved.

The proposed solution should not be completely ignored because it can reveal further significant information. Examining the estimated concentration factors closely, it can be seen that only the two actual components of the mixture have factors different from zero. Apparently zero concentration factor denies the presence of the reference compound in the mixture.

Furthermore in Table VII, it can be seen that the estimated concentration factors for the two components in Mixture 19 were not very far from the actual ones.

At this point however, since the solution from the NNLS method was not acceptable, the algorithm enters into the refining loop (Figure 19). The component that was found to have the largest concentration factor is subtracted from the unknown spectrum and the library search step will be

Table VII. Output of NNLS calculations for Mixture 19.

NON-NEGATIVE LEAST-SQUARES RESULTS FOR : MIXTURE 19

REFERENCE SPECTRUM	CONCENTRATION FACTOR	(ACTUAL)	
1) ca	9.46	(7.0)	*
2) da	0.00	(0.0)	
3) ph	3.02	(3.0)	*
4) tp	0.00	(0.0)	
5) di	0.00	(0.0)	
6) ta	0.00	(0.0)	

RNORM = 718474. MODE = 1

****

repeated.

In the case of our example, the HTL file of reference spectrum ca, which was found to have the largest concentration factor is subtracted from the HTL file of the unknown Mixture 19. Specifically, each point of the HTL file of reference ca, after been multiplied by the concentration factor (9.46) calcutated from the NNLS method, would be subtracted from the corresponding point of the HTL file of the unknown.

Because of the expected inaccuracy of the estimated concentration factor, to avoid situations where the reference spectrum is over-subtracted, the concentration factor is always rounded down before use. The factor 9.46 found for the reference spectrum ca, was rounded down to 9.0. This rule was applied in all cases, since in many instances the first estimated factors were found to be larger that the actual ones.

If the spectral subtraction was performed in the spectral domain were all data points would have a positive value, it would be possible to avoid over-subtraction of a compound by testing for the production of negative numbers. In the case of the present algorithm the subtraction is performed in the transform domain, where data points with both positive and negative values exist, such a test can no be used. The only method to avoid over-subtraction of a

compound was to round down the estimated concentration factors.

Following the diagram in Figure 19, the resulting new HTL file would be again clipped and the library would be re-searched against the new CLP file.

After the second library search, the reference spectra found to have the largest Positive/Negative Ratios will be checked against the reference spectra found to have zero concentration factors at the previous NNLS calculations. If a reference compound was found to have zero concentration, i.e. it could not be present in the unknown mixture, it will not re-enter at the NNLS method, even if it was found to have a large Positive/Negative Ratio during the second library search.

For the case of Mixture 19, the reference spectra that were selected during the second library search, along with the new concentration factors estimated during the second application of the NNLS method, are shown in Table VIII.

Comparing the new concentration factors, calculated-after the first iteration of the algorithm through the loop, with the actual factors shown in parenthesis, and also by observing the value of RNORM, one concludes that this is a very accurate and acceptable solution.

From the two examples above, it is apparent that the algorithm is able to correctly identify, and also very

Table VIII. Output of NNLS calculations for Mixture 19 after one iteration through the algorithm loop.

NON-NEGATIVE LEAST-SQUARES RESULTS FOR : MIXTURE 19

ITERATION : 1

REFERENCE SPECTRUM	CONCENTRATION FACTOR	(ACTUAL)	
1) ca	7.00	(7.0)	*
2) ph	3.00	(3.0)	*
3) ta	0.00	(0.0)	
4) sa	0.00	(0.0)	
5) pe	3.00	(3.0)	*
6) tq	0.00	(0.0)	

RNORM = 27. MODE = 1

****

accurately quantify, the components of these two mixtures.

The complete analysis of the remaining unknowns, are summarized in Table IX. It can be seen that the algorithm was able to completely and correctly analyze 92% of the unknowns tested (37 out of 40 unknowns). For the remaining unknowns the algorithm was still able to correctly identify and quantify some of the compounds present in the mixtures. Including these compounds, a 94% success rate was accomplished.

The analysis of those unknowns was usually achieved part either after the first run of the NNLS method or during the first iteration through the loop of the algorithm. Only about 20% of the trials required that the algorithm proceed into the loop two or three times. For the three cases where a complete analysis was not reached after the fourth time through the loop, the execution of the algorithm was terminated. In these three cases the great overlap of the components, as well the great resemblance of the actual components with other members of the reference library, made the analysis of the mixture impossible.

As can be seen in Table IX, for the 10 unknowns were artificial white noise was added into the spectra to test the behavior of the algorithm against noise, the procedure still gave excellent results. The presence of white noise did not affect the performance of the algorithm. The

Table IX. Summary of the results from the complete analysis of the 40 unknowns.

	Number of mixtures correctly analyzed	(%)	Number of compounds correctly quantified	(%)
Unknowns 1 to 10 (one component)	10	(100)	10	(100)
Mixtures 1 to 10 (two components)	10	(100)	20	(100)
Mixtures 11 to 20 (three components)	9	(90)	28	(93)
Mixtures 21 to 30 (five components)	8	(80)	45	(90)
TOTAL	27	(92)	103	(94)

behavior of the algorithm toward noise will further be examined in the next section.

After this thorough investigation of the library search, and the NNLS method parts of the algorithm using simulated mixtures, the next section will test the algorithm against actual EEM of mixtures of poly-aromatic hydrocarbons.

Before we proceed to the next section it is necessary to address a philosophical concern about the soundness of the presented chemometrics approach in contrast to more computer intensive approach.

First, the fact that the computing speed of today's computers has reached tremendous levels should not be confused with the limitations of the mathematical techniques that are implemented via these computers. The limitations of mathematical techniques are often not imposed by the speed of the computer or the person using the computer but by the very nature of the mathematical procedures and methods involved. The fault, dear Brutus, lies not in our "Computers", but ourselves. For example, the calculation of the square root of a negative number cannot be done even with the use of the fastest computer.

In some other situations, the limitations of the mathematical techniques used are imposed from the physical or practical meaning of the solutions that those techniques

offer. For example, if a simple Least Squares method (not the Non-Negative Least Squares method) is used to estimate the concentrations of the components of a mixture a mathematically acceptable solution can be found which shows the concentration of some of the components to be negative. Obviously such a solution would have no physical meaning.

In terms of the Least Square method used in this study, although theoretically the number of unknown parameters that can be estimated by the method could equal the number of data points in the measurement matrices, there would be no practical value in such a solution. As the authors of the book "*Solving Least Squares Problems*[74]" explain "...the purpose of least squares computation is not merely to find some set of numbers that 'solve' the problem, but rather the investigator wishes to obtain additional quantitative information describing the relationship of the solution parameters to the data..."

Forces solutions for complex systems can be extracted from shear application of computer time, but they usually have little practical meaning, and may be misleading. For example M.R. Thompson[78] has shown that correlation techniques can solve quantitative decomposition of Infrared spectra with up to 20 components but the discrepancies between calculated and known values are gross.

A final issue concernign the algorithm was investigated: how the algorithm reacts if a particular component of an unknown mixture is not a member of the reference library.

To test the algorithm in this situation, the analysis of some of the two and three component unknown mixtures was repeated after one of the components of those mixtures was removed from the reference library. The selection of the unknown mixtures to be used in this test, as well as which component of the mixture to be removed from the library was done randomly, with the use of the random number generator discussed previously.

Six two component mixtures and four three component mixtures were selected to be used. For some of those mixtures the component that was removed from the library, was a major component providing the main contribution to the fluorescence intensity of the unknown, while for some other mixtures the component removed was a minor component providing a small contribution to the fluorescence intensity of the mixture.

The result of this analysis was that the algorithm was not able to find an acceptable solution for any of these unknowns. The algorithm entered the loop four times (the maximum number of times allowed) but it was not able to find an acceptable solution. It should be noted that during the

previous analysis, the algorithm was able to correctly analyze all those ten unknown mixtures.

The result of this test on the behavior of the algorithm was expected and it gives greater confidence to the answers that the algorithm produces. The algorithm will not force a solution to the problem by giving an acceptable answer when such an answer is not available. If the algorithm is to be used under real circumstances it is preferable to not get a solution than to get an incorrect solution.

The unknown mixtures used in this test along with the component that was removed from the library are shown in the next page, 159A.

C. REAL MIXTURES

In this second part of the study, actual Excitation Emission Matrices of mixtures of poly-aromatic hydrocarbons in aqueous solutions at very low concentrations were used to test the performance of the algorithm.

The algorithm was employed to analyze this series of unknown mixtures, using a small reference library that was developed specifically for that purpose. Finally, the effect that chemical and electronic (white) noise would have on the performance of the algorithm was examined.

Mixture Number	Component removed (concentration factor)
1	bb (6.0)
3	py (1.0)
5	tm (8.0)
6	va (2.0)
7	id (3.0)
10	qp (7.0)
12	pd (3.0)
14	pa (6.0)
18	pd (1.0)
19	ca (9.0)

1. EXPERIMENTAL

Actual mixtures of poly-aromatic hydrocarbons were used to test the performance of the algorithm in this part of the study.

The instrument used to collect the Excitation Emission Matrices of the mixtures and the pure components was again the same Perkin-Elmer Fluorescence Spectrophotometer, Model MPF-66, which provides a 0.25nm to 20nm resolution range in 0.1nm increments (MPF-66 Operating Directions, Perkin-Elmer 1984). The instrument was connected to a Perkin-Elmer 7500 Professional Computer for data collection, and instrument control.

The settings of the instrument parameters used to collect the spectra for this part of the study are summarized in Table X. The selection of a 4nm resolution for the collected spectra, was a trade off between the scan width and resolution.

The use of much larger wavelength ranges would require a lower instrument resolution which would deteriorate the characteristics of the spectra. On the other hand, if a higher instrument resolution was selected the excitation and emission wavelength ranges would be smaller, resulting in less selective measurements. The chosen resolution was a trade off between selectivity, size of Excitation Emission Matrices, and computations time.

Table X. Settings of Instrumental Parameters.

Perkin-Elmer
Fluorescence Spectrophotometer
Model MPF-66

Excitation wavelength range	200nm to 452nm
Emission wavelength range	236nm to 488nm
Excitation monochrometer slit width	5nm
Emission monochrometer slit width	5nm
Instrument Resolution used	4nm
Wavelength scan speed	160nm/min
Signal amplifier gain	HIGH

The solutions of the pure compounds used to develop the reference library (Table XI), as well as the solutions of the mixtures which were prepared to be analyzed with the algorithm, were prepared in degassed, deionized water. The selection of water as the solvent was done because the algorithm was intended to be used as an easy and fast method for the detection of pollution in environmental samples, mainly water samples.

The selection of deionized water was done in order to avoid the presence of heavy metals in the solutions which would effect the fluorescence of the compounds by quenching.[1] The deionized water was degassed by keeping the water at boiling point for at least 15 minutes. The water was degassed in order to avoid the effect of quenching from dissolved oxygen.[1]

The concentrations of the pure compounds in the reference solutions as well as the concentrations of the components of the mixtures were kept at very low levels, in most cases bellow 10^{-6} M, in order to avoid the effect of self absorption, which appears at higher concentrations.[1] More details on the selection of the concentration range for the compounds in the reference library will be presented in the next section.

The solutions of the reference compounds were prepared

by initially dissolving a few milligrams of the compound in water. By appropriate dilutions the reference solution was brought to the desired concentration. The source and purity of the compounds used to develop the reference library as well as their exact concentrations can be found in Table XI.

The solutions of the unknown mixtures were prepared by combining portions of the reference solutions. By appropriate dilutions the solutions were brought to the desired concentrations. The selection of compounds to be used in the preparation of the unknown mixtures, as well as the concentrations of the reference compounds in the mixtures, was done with a random drawing with the use of a random number generator (see Experimental section on Simulated Mixtures).

By using Excitation Emission Matrices of solutions of actual mixtures, the behavior of the algorithm in the presence of instrumental as well as chemical noise (chemical interactions between species present in the mixtures) can be investigated.

2. POLY-AROMATIC HYDROCARBON MIXTURES

Poly-aromatic hydrocarbons, PAHs, with several aromatic benzenoid rings, are the most common type of molecules studied with the aid of fluorescence spectroscopy.

For the above reason, as well as the severe environmental hazard they present even in very low concentrations, PAHs were chosen for this study.

Before the development of the reference spectral library, some additional very important issues had to be investigated. The first issue is the concentration of the reference compounds in the solutions to be used to collect the EEMs for the reference library. The second issue is the verification of the assumption that the EEM of a mixture would be equal to the sum of the individual components. Finally, the effect of the presence of spectral background on the results of the algorithm needed to be investigated.

First, to find the optimum concentration for the reference solutions, an important assumption made by the algorithm, one discussed in the theory section, has to be recalled.

The algorithm assumes that the fluorescence intensity at each and every point of the EEM would change linearly with the concentration. The concentration of the reference solution, as well as the concentration of the reference compound in the unknown mixtures, would have to be in the linear part of the fluorescence intensity versus concentration relationship. The working range of the algorithm would be the range between L_{lower} and L_{upper}

(Figure 6).

To determine the working range, the EEM of anthranilic acid, one of the compounds to be used in the reference library, was collected in a number of different concentrations. The fluorescence intensity at the peak of the spectrum, as well as the intensity at half height, was plotted against the molar concentration, Figure 33.

From the plot in Figure 33 it can be seen that the fluorescence intensity remains linear with concentration, R^2=0.99, for a range of about three orders of magnitude, 10^{-8} to 10^{-5} M. The optimum concentration for the reference spectrum should be somewhere at the midpoint of the range. To obtain the best results, the same rule should be use for every compound that is member of the reference library.

To explore the second issue mentioned above, the linear additivity of the EEMs of the components in a mixture, the EEM of a mixture of two components, as well as the EEMs of the individual components were collected. In every case, the concentrations of the compounds were within the working range, as that was defined previously.

The sum of the EEMs of the two components was compared with the actual EEM of the mixture. The comparison can be seen in Figure 34. It is obvious that the EEMs of the individual components indeed add linearly to form the EEM of

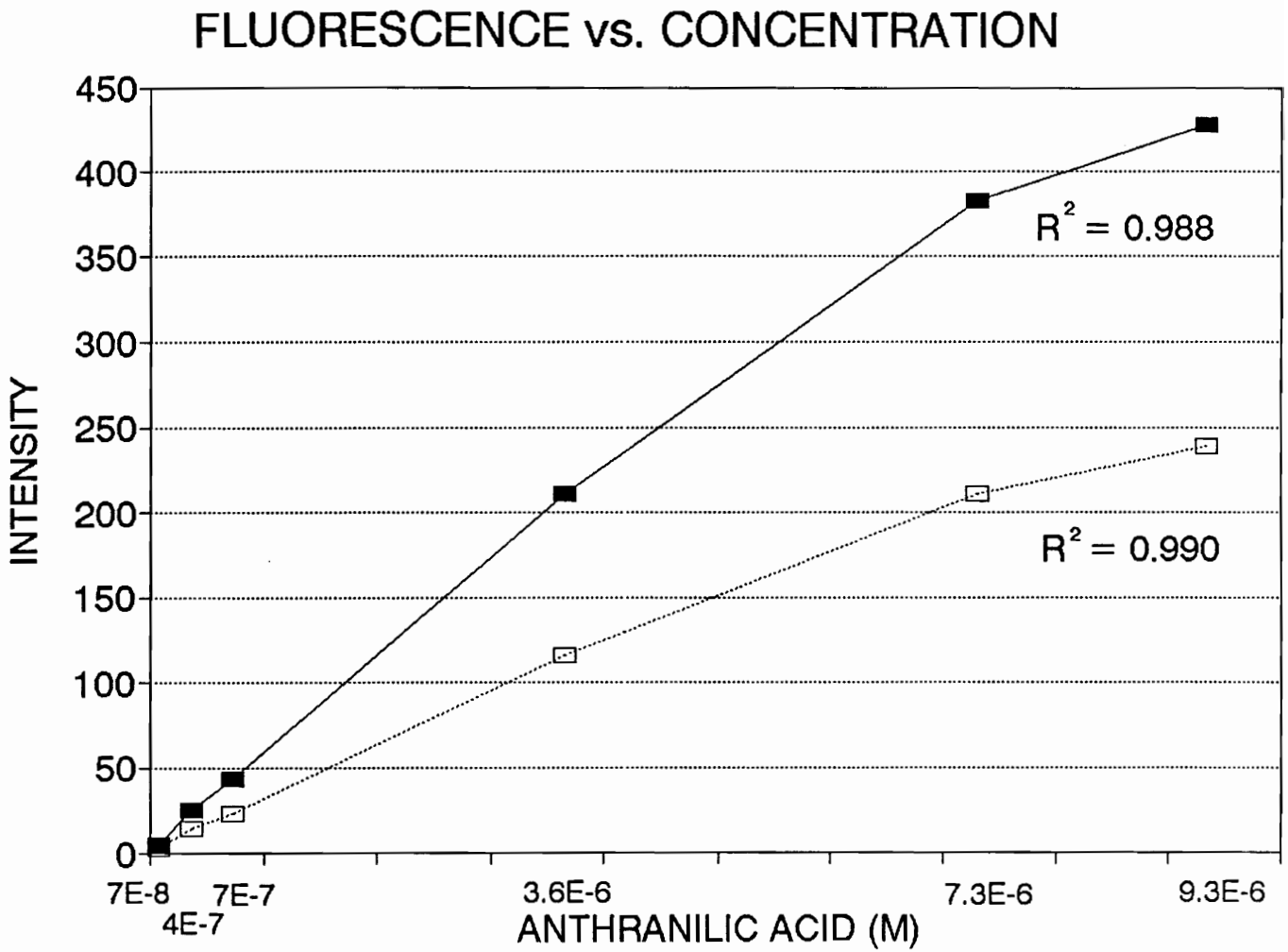


Figure 33. Working range for Anthranilic Acid.

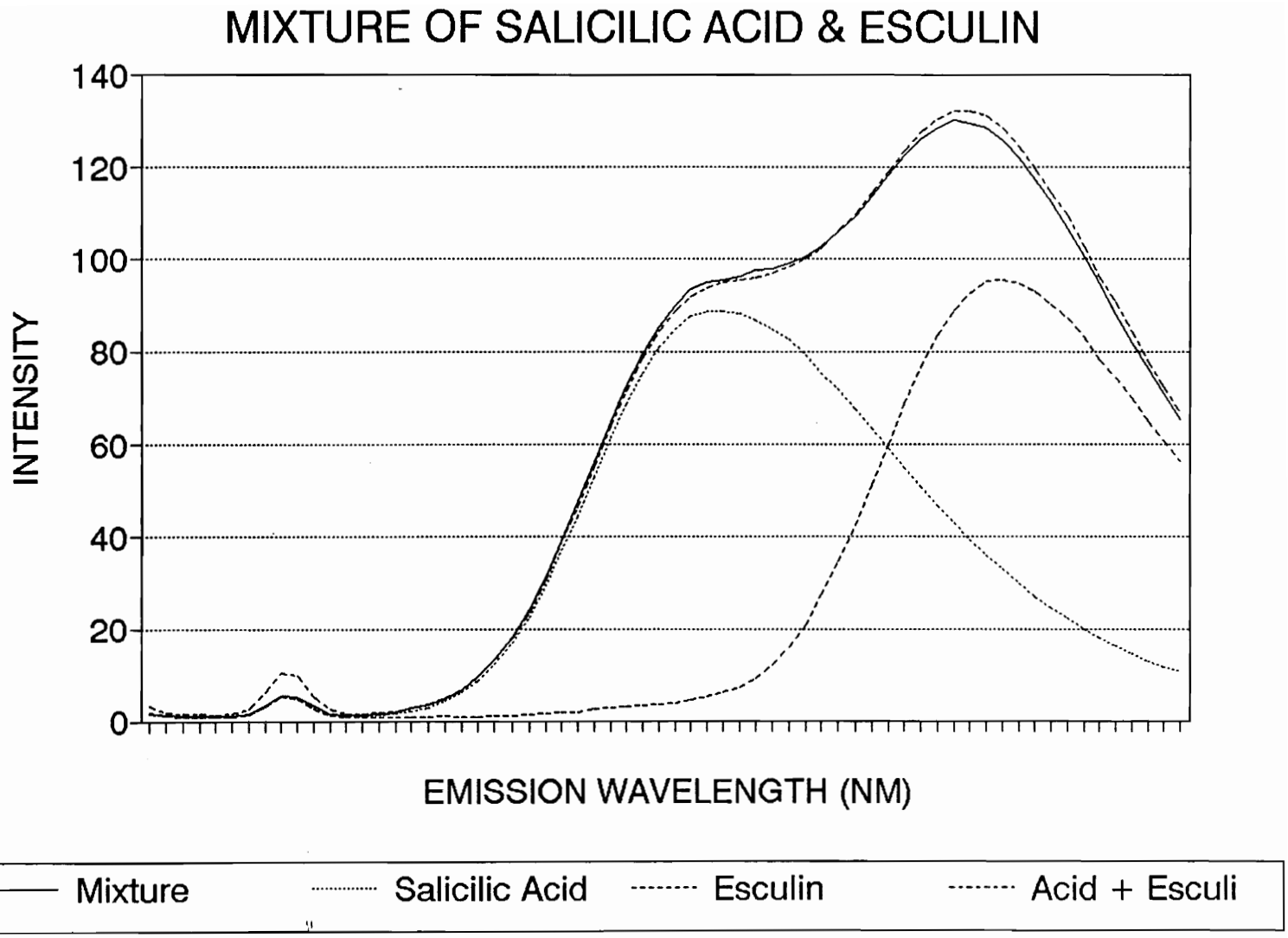


Figure 34. Sum of EEMs in a two component mixture.

the mixture within the working range of the algorithm.

To realize the importance of the third and last issue mentioned above, the background intensity, the EEM of the background has to be examined. The Excitation Emission Matrix of pure water, under the exact same conditions used for the other solutions, was collected to serve as the background. It is shown in Figure 35.

The background EEM consist of a large ridge of scattered light. This is a characteristic of the instrument design, not the result of the experimental technique used in this study. Each point on the ridge corresponds to an emission wavelength identical to the excitation wavelength used. Part of that scattered light is due to reflections on the instrument optics and the quartz cell walls. Another part comes from Rayleigh (elastic) scattering of the light from the water molecules. The intensity of Rayleigh scattered light is proportional to $1/\lambda^4$, which explains why the intensity of the scattered light dramatically decreases at higher excitation wavelengths, see Figure 35.

The background EEM also consists of a second low intensity ridge that runs almost parallel to that of the scattered light. This second ridge is formed from the Raman (inelastic) scattering of light. Because of the inelastic type of the phenomenon, i.e. energy loss is associated with

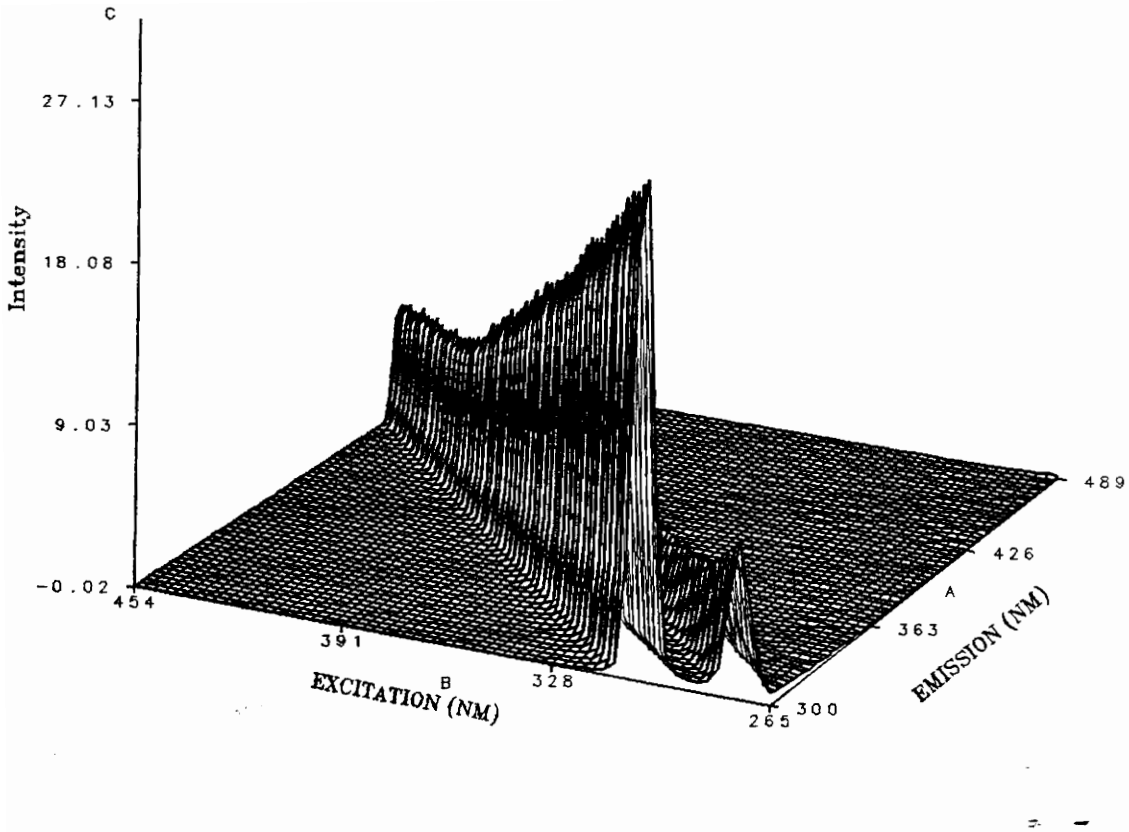


Figure 35. Background Excitation Emission Matrix.

the phenomenon, the emitted light appears at higher wavelengths compared to the wavelengths of the excitation beam.

The intensity of the Raman scattered light is often used to determine the sensitivity of fluorescence spectrophotometer. The signal to noise ratio for several points on that ridge, which correspond to vibrations of the water dipole at specific frequencies, can determine the sensitivity of the instrument. This method is preferred over the classic Limit of Detection method because it is simpler and it also can easily be employed to compare the sensitivity of different instruments.

The suggested value of signal to noise ratio determined with the above method for a well calibrated and maintained spectrophotometer should exceed a 30:1 ratio. Periodical testing of the fluorescence spectrophotometer employed in the present study gave a signal to noise ratio higher than 35:1.

Returning to the issue of the effect of the background on the performance of the algorithm, the intensity of the points of the EEM which contain background signal would no longer be linear with the concentration of the analyte. That would obviously effect the value of the saved Hartley transform points, the clipping pattern, and thus the results of the algorithm. The background signal has to be removed

before any further processing of the reference and unknown EEMs.

Because of the relative narrow shape of the scattered light ridge compared to the resolution with which the EEMs were collected, very small instrument fluctuations would produce large variations in the measured intensity of the scattered light ridge points. Thus a simple subtraction of the background signal would not be very reliable. An interactive method, where the user supervises the amount of subtracted signal, was used instead to remove the background signal.

Now that the above issues about the reference spectra in the library have been addressed, a reference library can be developed to help in the evaluation of the performance of the algorithm against EEMs of actual mixtures.

The Excitation Emission Matrices of several water soluble PAHs were collected to form a small reference library. The EEMs of the compounds in the reference library along with their structures can be seen in Appendix D. At the same time, several of those compounds were randomly selected to be used as unknowns. At this time actual solutions of those compounds were combined to serve as unknown mixtures.

Because of the small size of the reference library, only three unknown solutions were prepared. The three

unknown mixtures contained one, two, and three components, respectively. The composition of those unknown mixtures, and the names and concentrations of the compounds in the reference library, are shown in Table XI. The Excitation Emission Matrices of the three unknown solutions are shown in Figure 36, Figure 37, and Figure 38.

Following the results described earlier for the development of the reference library, the Excitation Emission Matrices of the compounds were unfolded and clipped, using the boustrophedon unfolding and relative clipping method C. The HTL and the CLP files were again stored to form the reference library.

The EEMs of the unknown mixtures were also unfolded, transformed, clipped, and the reference library was searched. The Positive/Negative Ratio for every member of the library was calculated and the compounds with Ratios higher than 1.5 were selected to enter in the NNLS calculations.

Because the unknown mixtures contained only a maximum of three components, and also because of the small size of the reference library, the minimum number of compounds to enter the NNLS calculations regardless of the Positive /Negative Ratio was set to four. The results of the reverse library search and the NNLS calculations for the three unknown mixtures are shown in Table XII.

Table XI. List of compounds in the reference library, and composition of the unknown mixtures.

KA	Salicylic Acid (Aldrich 99+%, Gold label)	2.2×10^{-6} M
KB	1,4-Naphthoquinone (Aldrich 97%)	1.0×10^{-4} M
KE	Linuron (EPA Standard)	4.0×10^{-6} M
KF	Indole (Aldrich 99+%, Gold label)	2.4×10^{-6} M
KH	Anthranilic Acid (Fluka AG, puriss.p.a)	6.6×10^{-7} M
KK	Fluorene (Aldrich 98%)	1.2×10^{-6} M
KP	1,4-Dimethoxybenzene (Aldrich 99%)	1.4×10^{-6} M
KT	Benzoquinone (Aldrich 98%)	1.4×10^{-5} M
KZ	Acridine (Aldrich 98%)	1.4×10^{-7} M
PU	PPD (Fluka AG, >98% purum)	3.8×10^{-8} M
PY	Esculin (Fluka AG, >98% purum)	8.8×10^{-8} M
Unknown 1.	Indole	2.4×10^{-6} M
Unknown 2.	Anthranilic Acid	6.6×10^{-7} M
	Indole	3.0×10^{-6} M
Unknown 3.	Linuron	3.0×10^{-6} M
	Anthranilic Acid	3.3×10^{-7} M
	Esculin	8.8×10^{-0} M

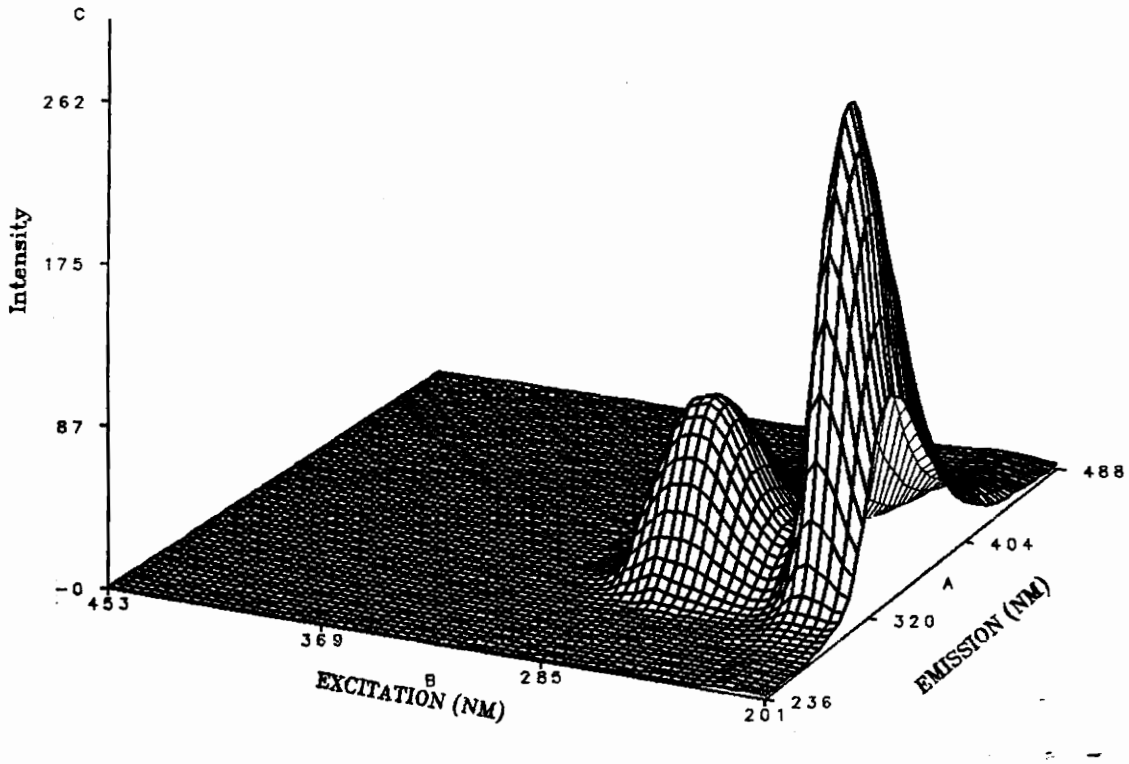


Figure 36. EEM of Unknown 1. (one component)

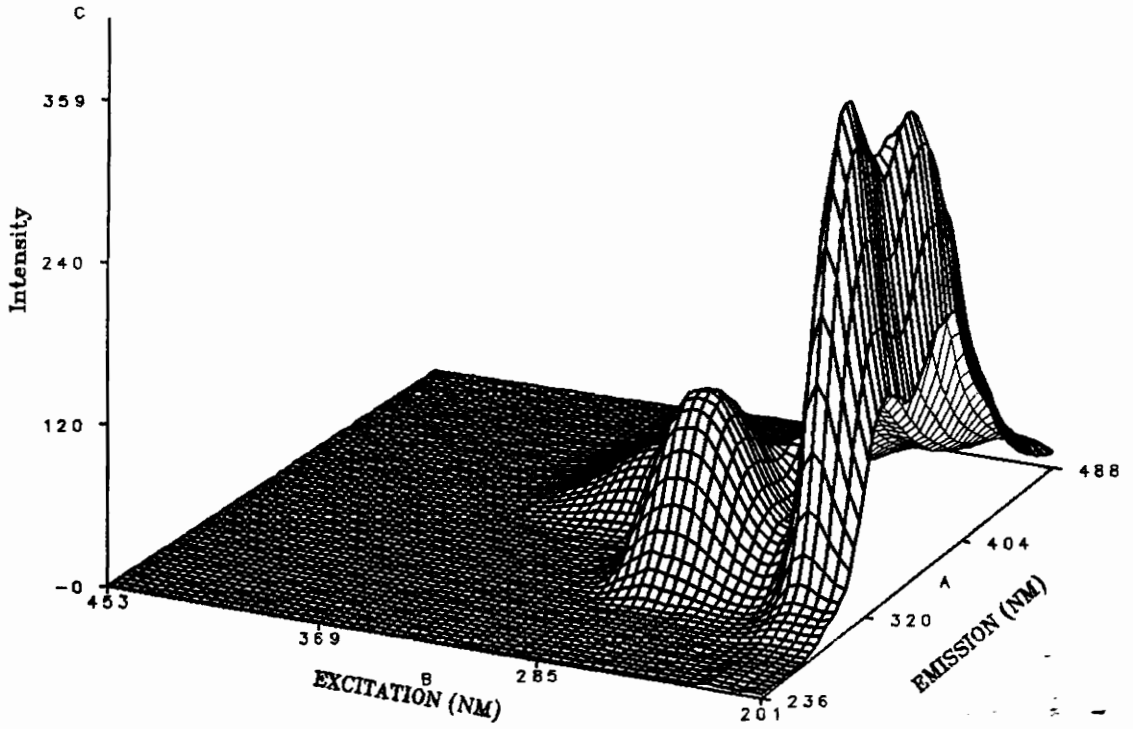


Figure 37. EEM of Unknown 2. (two components)

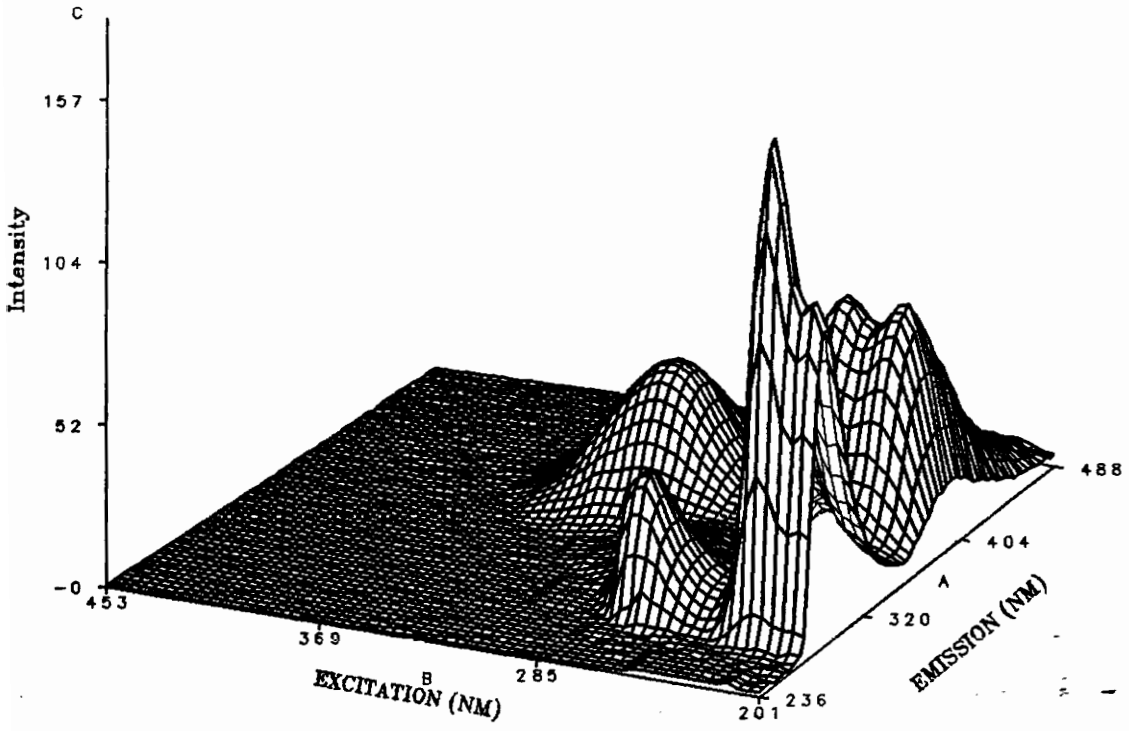


Figure 38. EEM of Unknown 3. (three components)

Table XII. Library search results and NNLS calculations for the three unknown mixtures.

NON-NEGATIVE LEAST-SQUARES RESULTS FOR : UNKNOWN 1.

REFERENCE SPECTRUM	CONCENTRATION FACTOR (ACTUAL)	
1) kf	1.0 (1.0)	*
2) pu	0.0 (0.0)	
3) kp	0.0 (0.0)	
4) kq	0.0 (0.0)	

RNORM = 120736. MODE = 1

****

NON-NEGATIVE LEAST-SQUARES RESULTS FOR : UNKNOWN 2.

REFERENCE SPECTRUM	CONCENTRATION FACTOR (ACTUAL)	
1) kh	1.0 (1.0)	*
2) kf	1.3 (1.25)	*
3) pu	0.1 (0.0)	
4) ka	0.0 (0.0)	

RNORM = 159139. MODE = 1

****

NON-NEGATIVE LEAST-SQUARES RESULTS FOR : UNKNOWN 3.

REFERENCE SPECTRUM	CONCENTRATION FACTOR (ACTUAL)	
1) ke	0.8 (0.75)	*
2) kh	0.5 (0.5)	*
3) kq	0.0 (0.0)	
4) py	0.1 (0.1)	*

RNORM = 228595. MODE = 1

In Table XII the four selected reference spectra, for each unknown, are ranked from highest to lowest Positive/Negative Ratio. Following the notation used in the previous part, the actual concentration factors are shown in parenthesis, ant the correct components of the unknowns are denoted with an asterisk. It can be seen that again the reference with the highest Ratio was always an actual component of the unknown.

The higher values of the RNORM, relative to those calculated during the use of simulated mixtures is due to the noise present in the EEMs and to instrument instability during the different runs. To verify this conclusion, the calculations were repeated after intentionally removing some of the actual components present in the mixtures. In other words, the method was forced to produce an non-acceptable solution, so that the value of the RNORM can be compaired with that of an acceptable solution. In each case the resulting RNORM was at least one order of magnitude larger, denoting a non-acceptable solution.

From Table XII it is apparent that the algorithm was able to successfully identify and quantify all the components of the three unknown mixtures. The fact that the analysis was completed during the first NNLS calculations and no further iterations through the loop of the algorithm were required, was attributed to the relative small size of

the library.

The importance of the obtained results, though, even with the small size library was still quite significant. The actual components of the unknown mixtures indeed had the highest calculated Positive/Negative Ratio in the library, and also the NNLS method was proven capable of correctly quantifying the components of real unknown solutions.

It should be further noted that the calculated concentration factors have only one significant figure after the decimal place because of the errors associated with the preparation of the reference and unknown solutions. Those errors were introduced from the instrumentation and apparatus involved during the preparation of the solutions, e.g. analytical balance, volumetric flasks, pipets, etc.

The last test that the algorithm was put through, since it was proven capable of dealing with real EEMs of actual multi-component mixtures, was the presence of chemical noise, i.e. chemical interactions between species present in the same unknown solution, as well as the present of white noise. Those last two issues are discussed in the next section.

2. CHEMICAL AND WHITE NOISE

In the case of multi-component mixture the possibility of some type of interaction between different species

present in the solution always exists. For example, those interactions include interactions between acids and bases, or interactions where two species interact to form some type of complex as in the case of charge transfer complexes.

In this part of the study, the performance of the presented algorithm was tested under the presence of acid-base and charge transfer interactions between the species of the unknown mixture.

The ability of the algorithm to deal with those situations was tested by selecting members of the developed reference library that would interact with each other to form unknown mixtures.

First, an acid (Anthranilic acid) and a base (Acridine) were selected and mixed together to form an unknown solution. Next, from the reference library two compound that could form a charge transfer complex, an electron donor (1,4-Dímethoxybenzene) and an electron acceptor (Benzoquinone), were selected and used to form a second unknown solution. The concentrations of the components in both solution were made to match the concentrations of the respective compounds in the reference library.

The Excitation Emission Matrices of the two solutions (Figure 39 and Figure 40) were collected and analyzed according to the algorithm. The results of the library search and the NNLS calculations are shown in Table XIII.

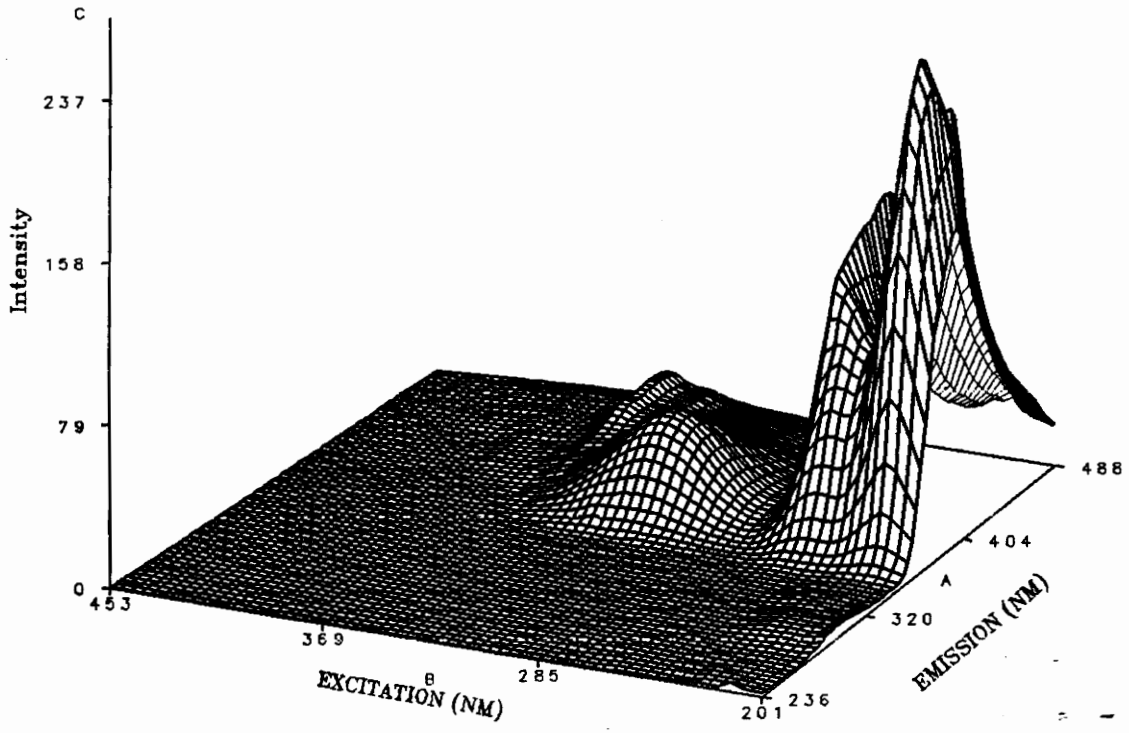


Figure 39. Anthranilic acid and Acridine.

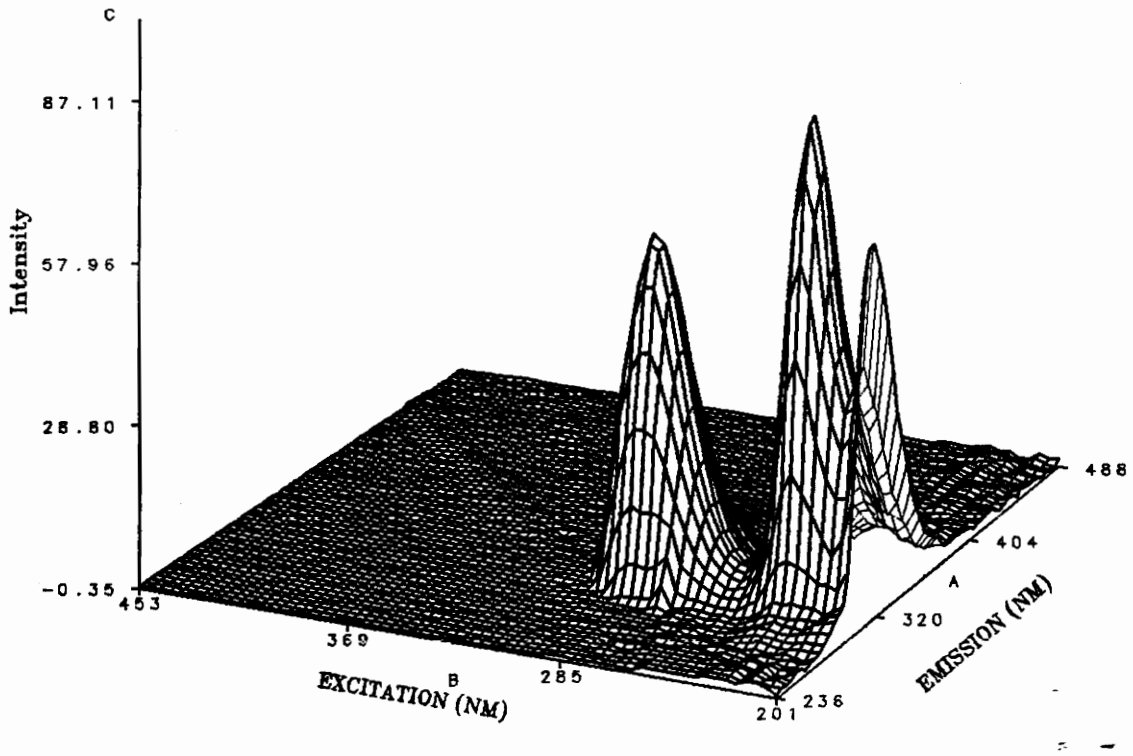


Figure 40. 1,4-Dimethoxybenzene and Benzoquinone.

Table XIII. NNLS results in the presence of acid-base and charge transfer complex interactions.

NNLS RESULTS FOR : ACID-BASE MIXTURE.

REFERENCE SPECTRUM	CONCENTRATION FACTOR (ACTUAL)	
1) kh	1.0 (1.0)	*
2) ka	0.0 (0.0)	
3) ky	1.0 (1.0)	*
4) kb	0.0 (0.0)	

RNORM = 82535 . MODE = 1

****

NNLS RESULTS FOR : CHARGE TRANSFER COMPLEX.

REFERENCE SPECTRUM	CONCENTRATION FACTOR (ACTUAL)	
1) kp	1.0 (1.0)	*
2) kq	0.9 (1.0)	*
3) pu	0.0 (0.0)	
4) kf	0.0 (0.0)	

RNORM = 18657. MODE = 1

****

Again, the reference compounds are shown in order of highest to lowest Positive/Negative Ratio, and the actual concentration factors are shown in parenthesis.

From Table XIII it is obvious that in both cases the algorithm was able to correctly identify the components of the unknown mixtures, and in the case of the acid-base was also able to accurately estimate the concentrations of the components.

In the case involving a charge transfer complex, the estimated concentration of the second component was found to be slightly lower than the actual one. The reason for that deviation was probably the fact that the contribution of that second component in the total fluorescence intensity of the mixture was relative small, and small instrument fluctuations were manifested in that manner.

From the above examples it appears that the above explained interactions do not depreciate the value of this decomposition approach. There are a number of reasons for this. First, the effect of quenching has been avoided by making sure that heavy metals and disolved oxygen are not present in the solutions. Second, the effect of acid base interactions between the components of the mixtures was reduced to minimum levels since the library containes only organic compounds. Organic acids and bases are mainly weak acids or bases.[79]

Finally, the effect of charge transfer formation between species present in the mixtures did not affect the results since the formation constants of those compounds are usually very small.[80]

At a first approximation the effects of quenching, acid base interactions, as well as charge transfer complex formation can be ignored. If the algorithm was to be extended to higher concentrations, outside the linear range of the fluorescence where self absorption can be observed, or strong acids and bases, or compounds that form strong charge transfer complexes are added in the library, then the algorithm would have difficulty. In those situations a different approach (e.g. neural networks) would have to be examined.

Finally, the performance of the algorithm was tested in the presence of significant white noise in the Excitation Emission Matrices of the unknown solutions.

For that reason, artificial white noise was added in the EEMs of the three unknown solutions used in the previous part of the study, the testing of the algorithm against actual mixtures. The noise was generated using a random number generator from a commercially available signal processing software package.

The same amount of noise, same peak to peak value, was added in the EEMs of each one of the mixture. The estimated

signal to noise ratios for the three unknowns were 27:1, 36:1, and 15:1 respectively. The resulting EEMs can be seen in Figure 41, Figure 42, Figure 43.

The EEMs were again unfolded, transformed, clipped, and the reference library was search. The new selected library members were entered in the NNLS calculations. The results of the library search, as well as the estimated factors from the NNLS method, are shown in Table XIV.

Examining the results in Table XIV, and comparing these results with those in Table XII it is apparent that the presence of the noise did not hinder the ability of the algorithm to analyze multi-component mixtures. Even in the case of the Unknown 3, which had a very small signal to noise ratio, the algorithm was still able to correctly identify and quantify all of the components. The algorithm also passed this last test successfully.

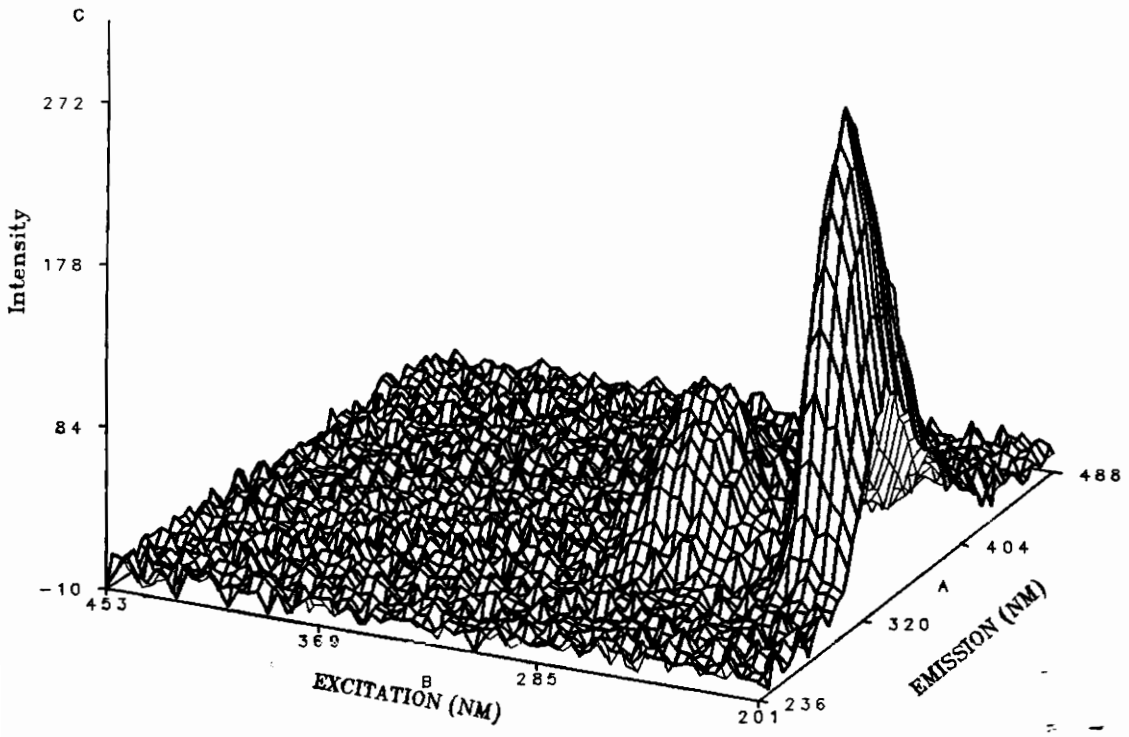


Figure 41. EEM of Unknown 1 with added white noise.

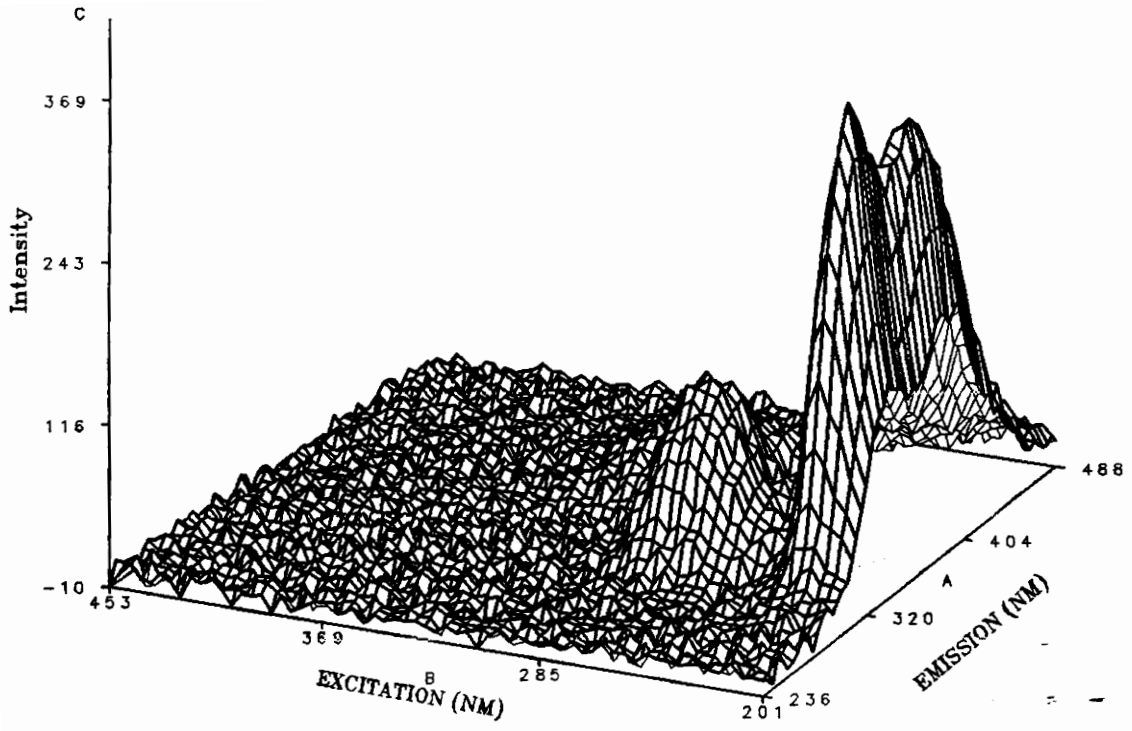


Figure 42. EEM of Unknown 2 with added white noise.

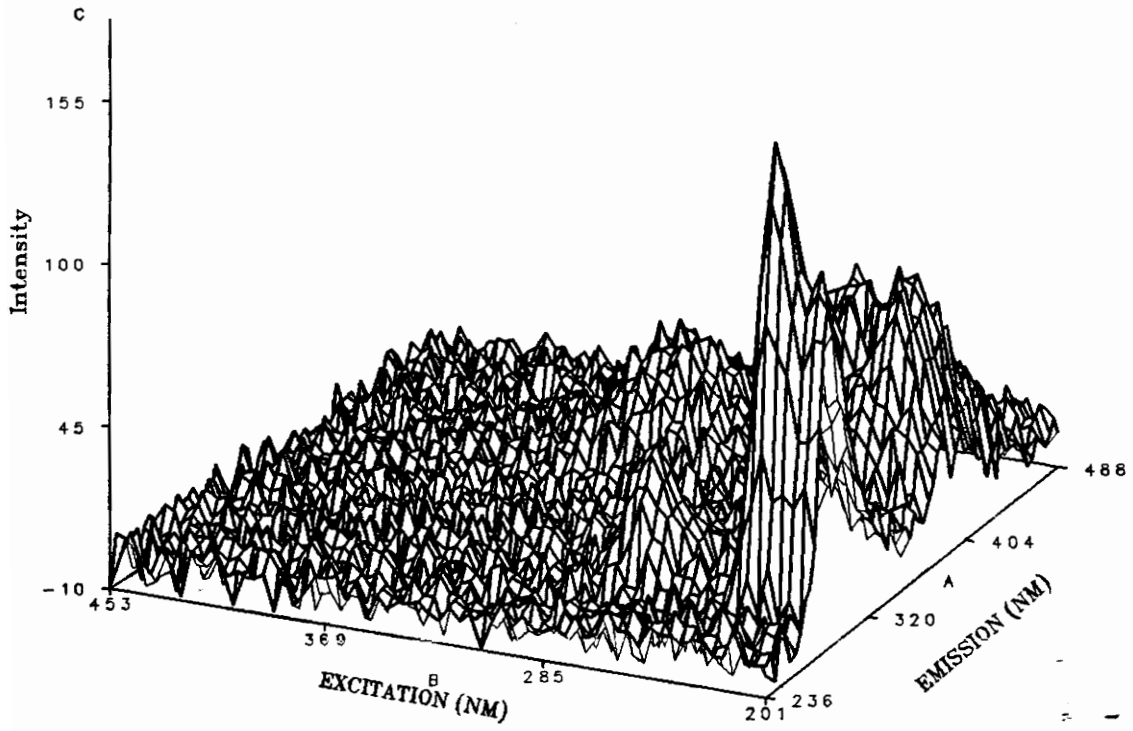


Figure 43. EEM of Unknown 3 with added white noise.

Table XIV. Results of the analysis of unknown real mixtures in the presence of white noise.

NON-NEGATIVE LEAST-SQUARES RESULTS FOR : UNKNOWN 1.

REFERENCE SPECTRUM	CONCENTRATION FACTOR	(ACTUAL)	
1) kf	1.0	(1.0)	*
2) pu	0.0	(0.0)	
3) kp	0.0	(0.0)	
4) ke	0.0	(0.0)	

RNORM = 123883. MODE = 1

****

NON-NEGATIVE LEAST-SQUARES RESULTS FOR : UNKNOWN 2.

REFERENCE SPECTRUM	CONCENTRATION FACTOR	(ACTUAL)	
1) kf	1.3	(1.25)	*
2) kh	1.0	(1.0)	*
3) ka	0.1	(0.0)	
4) kb	0.0	(0.0)	

RNORM = 162379. MODE = 1

****

NON-NEGATIVE LEAST-SQUARES RESULTS FOR : UNKNOWN 3.

REFERENCE SPECTRUM	CONCENTRATION FACTOR	(ACTUAL)	
1) ke	0.8	(0.75)	*
2) kh	0.5	(0.5)	*
3) ka	0.0	(0.0)	
4) py	0.1	(0.1)	*

RNORM = 232059. MODE = 1

VI. CONCLUSIONS

The goal of this research was the development of a Hartley transform based algorithm for the qualitative and quantitative analysis of multi-component mixtures using a compressed spectral library of Excitation Emission Matrices. The concept of a spectral library of pure compounds used for the analysis of multi-component mixtures was proved successful.

The developed algorithm proved capable of analyzing mixtures of five components with relative concentrations ratio of about 100:1 and significant spectral overlap. The algorithm, in 93% of the cases, was able to successfully identify the components in the mixtures and very accurately estimate their concentrations.

Also a number of techniques for pre-processing of three dimensional fluorescence spectra were investigated. The "boustrophedon" unfolding of three dimensional fluorescence spectra was one of the pre-processing techniques tested. The two-fold purpose of the unfolding was the reduction of the computations required to complete the Hartley transform of the spectra and at the same time the increase of the number of spatial frequencies that could be used for the identification of the components of a mixture.

The Hartley transform technique used in this research

is an alternative to the Fourier transform technique and a very powerful signal processing technique. The Hartley transform performs at least as well as the Fourier transform and at the same time it avoids the confusing concept of imaginary numbers.

The truncation of the Hartley transform spectrum along with the spectral encoding methods, especially relative encoding methods, was shown to be excellent compression tools for the Excitation Emission Matrices involved in this study. The spectra were compressed by a factor of almost 10:1 before they were included in the spectral library, while the most time consuming part of the algorithm, the library search, was greatly accelerated by been performed in a sub-set compressed by a factor 128:1 compared with the original data-set.

The Non-Negative Least-Squares method, employed in this study to estimate the concentrations of the components of the unknown mixtures, was shown to be a very powerful and robust computational method. The concept of non-negative coefficients agrees with the restriction of having only positive concentrations in chemical systems, making the method well suited for the analysis of chemical data.

Finally, the algorithm was also proved successful in the qualitative and quantitative analysis of mixtures when its performance was tested against the presence of large

amount of white noise in the spectrum of the unknown mixture, as well as the presence of chemical interactions between the species present in the unknown sample.

In conclusion, the research project undertaken was completed successfully, and the developed algorithm along with appropriate spectral libraries could be employed in a variety of real world situations, analysis of environmental samples, etc. The effect of chemical interferences and interactions between species present in the unknown solutions on the performance of the algorithm, as well as alternative methodologies for the compression and analysis of spectral data, e.g. neural networks, are areas where further research could very well be justified.

REFERENCES

1. J.R. Lakowicz, *Principles of Fluorescence Spectroscopy*, **1983**, Plenum Press, New York.

2. I.M. Warner, J.B. Callis, E.R. Davidson, M. Gouterman, G.D. Christian, *"Fluorescence Analysis : A new approach"*, *Anal. Lett.* **1975**, 8, 665-681.

3. J.B. Zung, R.L. Woodlee, M-R.S. Fuh, I.M. Warner, *SPIE Vol. 1054 Fluorescence Detection III*, **1989**, 69-76.

4. X.M. Tu, D.S. Burdick, D.W. Millican, L.B. McGown, *Anal. Chem.* **1989**, 61, 2219-2224.

5. Chou-Pong Pau, I.M. Warner, T.M. Rossi, *trend in analytical chemistry*, **1988**, 7, 68-73.

6. J.B. Zung, T.T. Ndou, I.M. Warner, *Anal. Chem.* **1990**, 44, 1491-1493.

7. T.M. Rossi, I.M. Warner, *Appl. Spectrosc.* **1985**, 39, 949-959.

8. C. Pau, G. Patonay, C.W. Moss, G.M. Carlone, T.M. Rossi, I.M. Warner, *Clin. Chem.* **1986**, 32/6, 987-991.

9. Chou-Pong Pau, I.M. Warner, *Appl. Spectrosc.* **1987**, 41, 496-502.

10. M. Vicsek, S.L. Neal, I.M. Warner, *Appl. Spectrosc.* **1986**, 40, 542-548.

11. T.M. Rossi, I.M. Warner, *Appl. Spectrosc.* **1984**, 38, 422-429.

12. I.M. Warner, G. Patonay, M.P. Thomas, *Anal. Chem.* **1985**, 57, 463-483A

13. J.R. Maple, E.L. Wehry, G. Mamantov, *Anal. Chem.* **1980**, 52, 920-924.

14. A.P. D'Silva, V.A. Fassel, *Anal. Chem.* **1984**, 56, 985-1000A.

15. E.V. Shpol'skii, *Pure Appl. Chem.* **1974**, 37, 183-195.

16. D.W. Millican, L.B. McGown, *Anal. Chem.* **1989**, 61, 580-583.

17. D.W. Millican, L.B. McGown, *Anal Chem.* **1990**, 62, 2242-2247.

18. P.M. Ritenour Hertz, L.B. McGown, *Appl. Spectrosc.* **1991**, 45, 73-79.

19. J.R. Lakowicz, R. Jayaweera, H. Szmacinski, W. Wiczk, *Anal. Chem.* **1990**, 62, 2005-2012.

20. F.V. Bright, *Appl. Spectrosc.* **1988**, 42, 1245-1250.

21. I.M. Warner, G.D. Christian, E.R. Davidson, J.B. Callis, *Anal. Chem.* **1977**, 49, 564-573.

22. I.M. Warner, S.L. Neal, T.M. Rossi, *Journal of Research of the National Bureau of Standards*, **1985**, 90, 487-493.

23. E.R. Malinowski, D.G. Honery, *Factor Analysis in Chemistry,* John Wiley, New York, **1980.**

24. B.G.M. Vandeginste, C. Sielhorst, M. Gerritsen, *trends in analytical chemistry,* **1988**, 7, 286-287.

25. M. Kubista, *Chemometrics and Intelligent Laboratory Systems,* **1990**, 7, 273-279.

26. M. McCue, E.R. Malinowski, *Anal. Chem. Acta,* **1981**, 133, 125-136.

27. T. Hirschfeld, *Anal. Chem.* **1976**, 48, 721-723.

28. M.P. Fogarty, I.M. Warner, *Anal. Chem.* **1981**, 53, 259-265.

29. M.P. Fogarty, I.M. Warner, *Appl. Spectrosc.* **1982**, 36, 460-466.

30. F. Dousseau, M. Therrien, M. Pezolet, *Appl. Spectrosc.* **1989**, 43, 538-542.

31. J.K Kauppinen, D.J. Moffatt, H.H. Mantsch, D.G. Cameron, *Appl. Spectrosc.* **1981**, 35, 271-276.

32. W.I. Friesen, K.H. Michaelian, *Appl. Spectrosc.* **1985**, 39, 484-490.

33. P.R. Griffiths, G.L. Pariente, *trends in analytical chemistry,* **1986**, 5, 209-215.

34. M.P. Fuller, G.L. Ritter, C.S. Draper, *Appl. Spectrosc.* **1988**, 42, 217-227.

35. M.P. Fuller, G.L. Ritter, C.S. Draper, *Appl. Spectrosc.* **1988**, 42, 228-236.

36. Y. Li-shi, S.P. Levine, *Anal. Chem.* **1989**, 61, 677-683.

37. M.B. Seasholtz, D.D. Archibald, A. Lorber, B.R. Kowalski, *Appl. Spectrosc.* **1989**, 43, 1067-1072.

38. I.M. Warner, E.R. Davidson, G.D. Christian, *Anal. Chem.* **1977**, 49, 2155-2159.

39. C.N. Ho, G.D. Christian, E.R. Davidson, *Anal. Chem.* **1980**, 52, 1071-1079.

40. C.-N. Ho, G.D. Christian, E.R. Davidson, *Anal. Chem.* **1981**, 53, 92-98.

41. E. Sanchez, B.R. Kowalski, *Anal. Chem.* **1986**, 58, 496-499.

42. B.E. Wilson, W.A. Lindberg, B.R. Kowalski, *J. Am. Chem. Soc.* **1989**, 111, 3797-3804.

43. B.E. Wilson, B.R. Kowalski, *Anal. Chem.* **1989**, 61, 2277-2284.

44. R.E. Kalman, *J. Basic Eng.* **1960**, 82, 35-45.

45. H.N.J. Poulisse, *Anal. Chim. Acta,* **1979**, 112, 361-374.

46. C.B.M. Didden, H.N.J. Poulisse, *Analytical Letters,* **1980**, 13, 921-935.

47. T.L. Cecil, S.C. Rutan, *Anal. Chem.* **1990**, 62, 1998-2004.

48. H.B. Woodruff, G.M. Smith, *Anal. Chem.* **1980**, 52, 2321-2327.

49. B.J. Wythoff, C.F. Buck, S.A. Tomellini, *Anal. Chim. Acta,* **1989**, 217, 203-216.

50. K. Funatsu, Y. Susuta, S. Sasaki, *J. Chem. Inf. Comput. Sci.* **1989**, 29, 6-11.

51. K. Jeb, V. Helbig, F. Greiner, J. Kaspareit, V. Rohde, T. Weirauch, *J. Quant. Spectrosc. Radiat. Transfer,* **1989**, 41, 67-78.

52. G.M. Pesyna, R. Venkataraghavan, H.G. Dayringer, F.W. McLafferty, *Anal. Chem.* **1976,** 48, 1362-1368.

53. B.L. Atwater, D.B. Stauffer, F.W. McLafferty, *Anal. Chem.* **1985,** 57, 899-903.

54. B.L. Atwater, R. Venkataraghavan, F.W. McLafferty, *Anal. Chem.* **1979**, 51, 1945-1949.

55. K.P. Cross, C.G. Enke, *Computers & Chemistry,* **1986**, 10, 175-181.

56. J.M. Bjerga, G.W. Small, *Anal. Chem.* **1990**, 62, 226-233.

57. O. Yamamoto, K. Hayamizu, M. Yanagisawa, *Analytical Sciences*, **1989**, 5, 141-146.

58. R.A. Divis, R.L. White, *Anal. Chem.* **1989**, 61, 33-37.

59. Z. Zolnai, S. Macura, J.L. Markley, *Journal of Magnetic Resonance*, **1988**, 80, 60-70.

60. F. Ishihara, "The use of Hadamard Transform as a data compression technique in the development of a 3-dimensional fluorescence spectral library for qualitative analysis", Ph.D. Dissertation, Virginia Polytechnic Institute and State University, **1988.**

61. J.I. Garrels, *The Journal of Biological Chemistry*, **1989**, 264, 5269-5282.

62. J.I. Garrels, B.R. Franza, Jr., *The Journal of Biological Chemistry*, **1989**, 264, 5283-5298.

63. I.E. Alguindigue, R.E. Uhrig, *Scientific Computing & Automation*. **1991**, 43-50.

64. J.B. Birks, *Photophysics of Aromatic Molecules*, **1970**, Wiley-Interscience, New York.

65. R.M. Miller, *Analytical Proceedings*, **1988**, 25, 350-354.

66. J.W. Cooley, J.W. Tukey, *Math. Comput.* **1965**, 19, 297-301.

67. R.N. Bracewell, *Proceedings of the IEEE*, **1984**, 72, 1010-1018.

68. R.V.L. Hartley, *Proceedings of the Inst. Radio Eng.* **1942**, 30, 144-150.

69. M.A. O'Neill, *B Y T E*, **1988**, 293-300.

70. R.N. Bracewell, *The Hartley Transform*, Oxford University Press, New York, **1986**.

71. P.R. Griffiths, *Transform Techniques in Chemistry*, Plenum Press, New York, **1978**.

72. G.C. Levy, S. Wang, P. Kumar, P. Borer, *Spectroscopy*, **1991**, 6, 20-33.

73. J.A. Storer, *Data Compression: Methods and Theory*, **1988**, Computer Science Press, Rockville, Maryland.

74. C.L. Lawson, R.J. Hanson, *Solving Least Squares Problems*, **1974**, Prentice-Hall, Englewood Cliffs, New Jersey.

75. L. Ott, *An Introduction to Statistical Methods and Data Analysis*, **1988**, PWS-Kent, Boston.

76. R.W. Farebrother, *Linear Least Squares Computations*, **1988**, Marcel Dekker, New York.

77. B. Lam, S.J. Foulk, T.L. Isenhour, *Anal. Chem.* **1981**, 53, 1679-1684.

78. M.R. Thompson, "Evaluation of an analog front end processor", Ph.D. Dissertation, Virginia Polytechnic Institute and State University, **1983**.

79. S.H. Pine, J.B. Hendrickson, D.J. Cram, G.S. Hammond, *Organic Chemistry*, **1980**, McGraw-Hill, New York.

80. M. Tamres, R.L. Strong, *Contact Charge-transfer Spectra*,

APPENDICES

Appendix A.
Listings of major programs.

```
program unfold (input,output);

(*************************************************************)
(*                                                           *)
(*   PROGRAM UNFOLD                                 3/5/90   *)
(*                                                           *)
(*      This program takes the 64x64 spectrum and it         *)
(*   "boustraphedon" unfolds it into a linear array of       *)
(*   4096 elements.  It asks for the spectrum/input file     *)
(*   name and also for the unfolded/output file name.        *)
(*                                                           *)
(*                                    George Asimopoulos     *)
(*                                                           *)
(*************************************************************)

var
   i, j, k             : integer ;

   point               : array [1..8,1..8] of integer ;

   input_file,
   output_file         : text ;

   spectrum_file,
   unfolded_file       : packed array[1..70] of char ;

begin
   write('Enter spectrum/input file  : ');
   readln(spectrum_file);
   write('Enter unfolded/output file : ');
   readln(unfolded_file);

   open(input_file,spectrum_file,old);
   reset(input_file);
   open(output_file,unfolded_file);
   rewrite(output_file);

   for k:=1 to 32 do
      begin
         for i:=1 to 8 do
            begin
               for j:=1 to 8 do
                  begin
                     read(input_file,point[i,j]);
                     write(output_file,point[i,j]);
                  end;
               writeln(output_file);
            end;
```

```
         for i:=1 to 8 do
            begin
               for j:=1 to 8 do
                  read(input_file,point[i,j]);
            end;

         for i:=8 downto 1 do
            begin
               for j:=8 downto 1 do
                  write(output_file,point[i,j]);
               writeln(output_file);
            end;
      end;
end.
```

```
program hartley (input,output);

(*************************************************************)
(*                                                           *)
(*    PROGRAM HARTLEY                          3/5/90        *)
(*                                                           *)
(*    Fast Hartley transform routine.                        *)
(*                                                           *)
(*    transform:= forwd ; from time to frequency domain.     *)
(*    transform:= revse ; from frequency to time.            *)
(*                                                           *)
(*    power_index:= index to which 2 must be raised to       *)
(*    generate a transform containing 'syze' elements.       *)
(*                                                           *)
(*    syze:= number of elements in the input data array.     *)
(*                                                           *)
(*       This program can calculate the hartley              *)
(*    transformation of any number of points that is a       *)
(*    power of 2. The program can also  calculate the        *)
(*    reverse transformation. The output is integers.        *)
(*       The program asks for the input and output file      *)
(*    name, the number of points and the direction of        *)
(*    the transformation.                                    *)
(*                                                           *)
(*                                  George Asimopoulos       *)
(*                                                           *)
(*************************************************************)

const
   datasize = 4096 ;
type
   direction_type = ( forwd, revse );
   data_array_type = array[1..datasize] of real ;
   name = packed array[1..70] of char ;

var
   dir, test_option, dummy : char ;
   i, j, k, syze, iter, power_index, demo : integer ;
   data_array : data_array_type ;
   transform_direction : direction_type ;
   spectrum_file, transform_file : name ;
   input_file, output_file : text ;

(*************************************************************)

procedure fht ( var data_array : data_array_type ;
                    power_index, syze : integer ;
                    transform_direction : direction_type ) ;
```

```
var
   i,j,k,
   trg_ind,trg_inc,
   power,t_a,f_a,
   i_temp,section,
   s_start,s_end : integer ;
   sne, csn : array[1..datasize] of real ;
   accu : array[1..2,1..datasize] of real ;

(**************************)

function permute ( index : integer ) : integer ;

var
   i, j, s : integer ;

begin
   j := 0 ;
   index := index - 1 ;
   for i:=1 to power_index do
      begin
         s := index div 2 ;
         j := j + j + index - s - s ;
         index := s ;
      end;
   permute := j + 1 ;
end;

(********************************)

procedure trig_table ( npts : integer ) ;

const
   pi = 3.14159265 ;
var
   i : integer ;
   angle, omega : real ;

begin
   angle := 0 ;
   omega := 2 * pi / npts ;
   for i:=1 to npts do
      begin
         sne[i] := sin(angle) ;
         csn[i] := cos(angle) ;
         angle := angle + omega ;
      end;
end;
```

```
(*****************************)

function modify ( power, s_start, s_end, index : integer ) :
integer ;

begin
   if ( s_start = index ) or ( power < 3 ) then
      modify := index
   else
      modify := s_start + s_end - index + 1 ;
end;

(*****************************)

procedure butterfly ( trig_ind, i_1, i_2, i_3 : integer ) ;

begin
   accu[t_a,i_1] := accu[f_a,i_1] +
                    accu[f_a,i_2] * csn[trig_ind] +
                    accu[f_a,i_3] * sne[trig_ind] ;
   trig_ind := trig_ind + syze div 2 ;
   accu[t_a,i_2] := accu[f_a,i_1] +
                    accu[f_a,i_2] * csn[trig_ind] +
                    accu[f_a,i_3] * sne[trig_ind] ;
end;

begin
   power := 1 ;
   f_a := 1 ;
   t_a := 2 ;
   trig_table(syze) ;
   for i:=1 to syze do
      begin
         accu[f_a,permute(i)] := data_array[i] ;
      end;
   for i:=1 to power_index do
      begin
         j := 1 ;
         section := 1 ;
         trg_inc := syze div (power + power) ;
         repeat
            trg_ind := 1 ;
            s_start := section * power + 1 ;
            s_end := ( section + 1 ) * power ;
            for k:=1 to power do
               begin
                  butterfly(trg_ind,j,j + power,
                            modify(power,s_start,s_end,j
                                                + power));
```

```
                    trg_ind := trg_ind + trg_inc ;
                    j := j + 1 ;
                end;
             j := j + power ;
             section := section + 2 ;
          until j > syze ;
          power := power + power ;
          i_temp := t_a ;
          t_a := f_a ;
          f_a := i_temp ;
       end;

    case transform_direction of
       forwd : for i:=1 to syze do
                    data_array[i] := accu[f_a,i] / syze ;
       revse : for i:=1 to syze do
                    data_array[i] := accu[f_a,i] ;
       end;
end;

(*************************************************************)

begin
   write('What is the name of the input file :  ');
   readln(spectrum_file);
   open(input_file, spectrum_file, old);
   reset(input_file);
   writeln;

   write('What is the name of the output file :  ');
   readln(transform_file);
   open(output_file, transform_file );
   rewrite(output_file);
   writeln;

   write('How many points you have : ');
   readln(syze);
   writeln;

   power_index := 0 ;
   demo := syze ;
   repeat
      demo := demo div 2 ;
      power_index := power_index + 1 ;
   until (demo = 1);

   writeln('Select transform direction :');
   write('          (F)orward, (R)everse  ');
   read(dummy);
```

```
   writeln;
   case dummy of
      'F','f' : transform_direction := forwd ;
      'R','r' : transform_direction := revse ;
   end;

   for i:=1 to syze do
       read(input_file, data_array[i]);

   fht( data_array, power_index, syze, transform_direction);

   i := 0 ;
   for k:=1 to 512 do
      begin
         for j:=1 to 8 do
            begin
               i := i + 1 ;
               write(output_file,round(data_array[i]):10);
            end;
         writeln(output_file);
      end;

   close(output_file);

end.
```

```
program hartley_2d (input,output);

(*************************************************************)
(*                                                           *)
(*    PROGRAM HARTLEY_2D                        10/17/90     *)
(*                                                           *)
(*    Fast Hartley transform routine.                        *)
(*                                                           *)
(*    transform:= forwd ; from time to frequency domain.     *)
(*    transform:= revse ; from frequency to time.            *)
(*                                                           *)
(*    power_index:= index to which 2 must be raised to       *)
(*                  generate a transform containing          *)
(*                  'syze' elements.                         *)
(*                                                           *)
(*    syze:= number of elements in the input data array.     *)
(*                                                           *)
(*    This program calculates the 2-D Hartley                *)
(*    transformation of a 64x64 data set. The program        *)
(*    can also calculate the reverse transformation. The     *)
(*    output is integers.                                    *)
(*    The program asks for the input and output file         *)
(*    names and the direction of the transformation.         *)
(*                                                           *)
(*                                   George Asimopoulos      *)
(*                                                           *)
(*************************************************************)

const
   datasize = 64 ;

type
   direction_type = ( forwd, revse );
   data_array_type = array[1..datasize] of real ;
   name = packed array[1..70] of char ;

var
   dir, test_option, dummy  : char ;
   i, j, k, syze, iter,
   power_index, demo        : integer ;
   spectrum                 : array[1..datasize,1..datasize]
                                                  of real ;
   data_array               : data_array_type ;
   transform_direction      : direction_type ;
   spectrum_file,
   transform_file           : name ;
   input_file, output_file  : text ;

(* ************************************************** *)
```

```
procedure fht ( var data_array : data_array_type ;
                    power_index, syze : integer ;
                    transform_direction : direction_type ) ;

var
   i,j,k,
   trg_ind,trg_inc,
   power,t_a,f_a,
   i_temp,section,
   s_start,s_end : integer ;
   sne, csn : array[1..datasize] of real ;
   accu : array[1..2,1..datasize] of real ;

(***************************)

function permute ( index : integer ) : integer ;

var
   i, j, s : integer ;

begin
   j := 0 ;
   index := index - 1 ;
   for i:=1 to power_index do
      begin
         s := index div 2 ;
         j := j + j + index - s - s ;
         index := s ;
      end;
   permute := j + 1 ;
end;

(***************************)

procedure trig_table ( npts : integer ) ;

const
   pi = 3.14159265 ;
var
   i : integer ;
   angle, omega : real ;

begin
   angle := 0 ;
   omega := 2 * pi / npts ;
   for i:=1 to npts do
      begin
         sne[i] := sin(angle) ;
         csn[i] := cos(angle) ;
```

```
         angle := angle + omega ;
      end;
end;

(*****************************)

function modify ( power, s_start, s_end, index : integer ) :
                  integer ;

begin
   if ( s_start = index ) or ( power < 3 ) then
      modify := index
   else
      modify := s_start + s_end - index + 1 ;
end;

(*****************************)

procedure butterfly ( trig_ind, i_1, i_2, i_3 : integer ) ;

begin
   accu[t_a,i_1] := accu[f_a,i_1] +
                    accu[f_a,i_2] * csn[trig_ind] +
                    accu[f_a,i_3] * sne[trig_ind] ;
   trig_ind := trig_ind + syze div 2 ;
   accu[t_a,i_2] := accu[f_a,i_1] +
                    accu[f_a,i_2] * csn[trig_ind] +
                    accu[f_a,i_3] * sne[trig_ind] ;
end;

begin
   power := 1 ;
   f_a := 1 ;
   t_a := 2 ;
   trig_table(syze) ;
   for i:=1 to syze do
      begin
         accu[f_a,permute(i)] := data_array[i] ;
      end;
   for i:=1 to power_index do
      begin
         j := 1 ;
         section := 1 ;
         trg_inc := syze div (power + power) ;
         repeat
            trg_ind := 1 ;
            s_start := section * power + 1 ;
            s_end := ( section + 1 ) * power ;
            for k:=1 to power do
```

```
                begin
                   butterfly(trg_ind,j,j + power,
                             modify(power,s_start,s_end,j +
                             power));
                   trg_ind := trg_ind + trg_inc ;
                   j := j + 1 ;
                end;
             j := j + power ;
             section := section + 2 ;
          until j > syze ;
          power := power + power ;
          i_temp := t_a ;
          t_a := f_a ;
          f_a := i_temp ;
       end;

   case transform_direction of
      forwd : for i:=1 to syze do
                   data_array[i] := accu[f_a,i] / syze ;
      revse : for i:=1 to syze do
                   data_array[i] := accu[f_a,i] ;
      end;
end;

(* ********************************************************** *)

begin
   write('What is the name of the input file :  ');
   readln(spectrum_file);
   open(input_file, spectrum_file, old);
   reset(input_file);

   write('What is the name of the output file :  ');
   readln(transform_file);
   open(output_file, transform_file );
   rewrite(output_file);

   for i:=1 to datasize do
      for j:=1 to datasize do
         read(input_file,spectrum[i,j]);

   syze := datasize ;

   power_index := 0 ;
   demo := syze ;
   repeat
      demo := demo div 2 ;
      power_index := power_index + 1 ;
   until (demo = 1);
```

```
writeln('Select transform direction :');
write('          (F)orward, (R)everse  ');
read(dummy);
writeln;
case dummy of
   'F','f' : transform_direction := forwd ;
   'R','r' : transform_direction := revse ;
end;

for i:=1 to datasize do
   begin
   for j:=1 to datasize do
      data_array[j] := spectrum[i,j];
   fht(data_array,power_index, syze,transform_direction);
   for j:=1 to datasize do
      spectrum[i,j] := data_array[j];
   end;

for j:=1 to datasize do
   begin
   for i:=1 to datasize do
      data_array[i] := spectrum[i,j];
   fht(data_array,power_index,syze,transform_direction);
   for i:=1 to datasize do
      spectrum[i,j] := data_array[i];
   end;

for i:=1 to 16 do
   begin
      k:=0;
      for j:=1 to 16 do
         begin
         write(output_file,round(spectrum[i,j]):10);
         k:=k+1;
         if (k=8)  then
            begin
            k:=0;
            writeln(output_file);
            writeln('i=',i:3,'  j=',j:3);
            end;
         end;
      k:=0;
      for j:=49 to 64 do
         begin
         write(output_file,round(spectrum[i,j]):10);
         k:=k+1;
         if (k=8)  then
            begin
            k:=0;
```

```
                writeln(output_file);
                end;
            end;
        end;
    for i:=49 to 64 do
        begin
            k:=0;
            for j:=1 to 16 do
                begin
                write(output_file,round(spectrum[i,j]):10);
                k:=k+1;
                if (k=8)  then
                    begin
                    k:=0;
                    writeln(output_file);
                    end;
                end;
            k:=0;
            for j:=49 to 64 do
                begin
                write(output_file,round(spectrum[i,j]):10);
                k:=k+1;
                if (k=8)  then
                    begin
                    k:=0;
                    writeln(output_file);
                    end;
                end;
        end;
    close(output_file);
end.
```

```
PROGRAM CLIPC_500 (INPUT,OUTPUT);

(*************************************************************)
(*                                                           *)
(*   PROGRAM CLIPC_500                          3/29/91      *)
(*                                                           *)
(*   This program is clipping the input file into 1's,       *)
(*   0's and -1's. The program compares three points at      *)
(*   a time and if the middle point is a maximum it clips    *)
(*   it into a 1, if it is a local minimum it clips it       *)
(*   into a -1, and the points in between if they are        *)
(*   closer to the maximum it clips them to 1 and if they    *)
(*   are closer to the minimum it clips them into -1. If     *)
(*   they are not close enough to either the maximum or      *)
(*   the minimum it clips them into 0.                       *)
(*   To check if they are close enough or not it takes       *)
(*   the difference between the min and max points and       *)
(*   25% of this difference is close to the minimum and      *)
(*   25% is close to the minimum. The rest of the            *)
(*   difference is considered as 0's. It clips only 512      *)
(*   points.                                                 *)
(*                                                           *)
(*                              George Asimopoulos           *)
(*                                                           *)
(*************************************************************)

TYPE
   name              = packed array[1..70] of char ;

VAR
   i, k, j, l,
   one, two, three,
   max_index, min_index,
   difference                : integer ;
   data_array,
   clipped_array             : array[1..512] of integer ;
   input_file,output_file    : text ;
   filtered_file,
   clipped_file              : name ;
   flag_max,flag_min         : boolean ;

begin
   write('What is the input file  : ');
   readln(filtered_file);
   open(input_file,filtered_file,old);
   reset(input_file);

   write('What is the output file : ');
   readln(clipped_file);
```

```
open(output_file,clipped_file);
rewrite(output_file);

for i:=1 to 512 do
   read(input_file,data_array[i]);
reset(input_file);

read(input_file,one,two);
clipped_array[1] := 1 ;
max_index := 1 ;
min_index := 2 ;
i := 1 ;

repeat
   i := i + 1 ;
   read(input_file,three);
   if (two>one) and (two>three)
      then
      begin
      max_index := i ;
      clipped_array[i] := 1 ;
      flag_max := true ;
      end;
   if (two<one) and (two<three)
      then
      begin
      min_index := i ;
      clipped_array[i] := -1 ;
      flag_min := true ;
      end;

   if (flag_min) and ((min_index - max_index)>1)
      then
      begin
      difference := data_array[max_index] -
                    data_array[min_index];
      for k:=(max_index+1) to (min_index-1) do
         begin
         clipped_array[k] := 0 ;
         if ( data_array[k] > ( data_array[max_index] -
                              ( 0.25 * difference ) ) )
            then
            clipped_array[k] := 1 ;
         if ( data_array[k] < ( data_array[min_index] +
                              ( 0.25*difference ) ) )
            then
            clipped_array[k] := -1 ;
         end;
      end;
```

```
      if (flag_max) and ((max_index - min_index)>1)
         then
         begin
         difference := data_array[max_index] -
                       data_array[min_index];
         for k:=(min_index+1) to (max_index-1) do
            begin
            clipped_array[k] := 0 ;
            if data_array[k] < ( data_array[min_index] +
                                 0.25 * difference )
               then
               clipped_array[k] := -1 ;
            if data_array[k] > ( data_array[max_index] -
                                 0.25 * difference )
               then
               clipped_array[k] := 1 ;
            end;
         end;

      flag_min := false ;
      flag_max := false ;
      one := two ;
      two := three ;

   until i=511 ;

   if (three>0)
      then
         clipped_array[512] := 1
      else
         clipped_array[512] := 0;

   i := 0 ;
   repeat
      for k:=1 to 16 do
            write(output_file,clipped_array[i+k]:3);
      i:=i+16;
      writeln(output_file);
   until i=512 ;

   close(input_file);
   close(output_file);

end.
```

```
program filter (input,output);

(*************************************************************)
(*                                                           *)
(*    PROGRAM FILTER                          5/29/91        *)
(*                                                           *)
(*    This program calculates the standard deviation of      *)
(*    the noise part of the transformation, it writes the    *)
(*    standard deviation and calculates the noise level:     *)
(*             noise = 1.96 * stdev (95% filter).            *)
(*    Then it checks each point in the transformed           *)
(*    spectrum against the noise level and if the point      *)
(*    is between 0.0 +/- noise it converts it into 0.        *)
(*    The number of points it uses for the calculation       *)
(*    of s is 1000 points.  The program asks for the         *)
(*    input and output file names, and it writes the         *)
(*    standard deviation.                                    *)
(*                                                           *)
(*                                    George Asimopoulos     *)
(*                                                           *)
(*************************************************************)

var
   i, j, k, number_of_points,
   index_number1, index_number2 : integer;

   Sum_of_X, Sum_of_X2, noise,
   stad_deviation, A, B, C      : real;

   data_array                   : array[1..4096] of real;

   input_file, output_file      : text;

   transformed_file,
   filtered_file                : packed array[1..70] of
                                                   char;


begin
   write('Enter the transformed/input file :  ');
   readln(transformed_file);
   open(input_file,transformed_file,history:=old);
   reset(input_file);

   for i:=1 to 4096 do
      read(input_file,data_array[i]);

   write('Enter the filtered/output file   :  ');
   readln(filtered_file);
```

```
   open(output_file,filtered_file);
   rewrite(output_file);

(* Calculate standard deviation ****************** *)

   number_of_points := 1000 ;

   index_number1 := 2048 - Round(number_of_points/2);
   index_number2 := 2048 + Round(number_of_points/2);

   Sum_of_X := 0 ;
   Sum_of_X2 := 0 ;

   for i:=index_number1 to index_number2 do
      begin
         Sum_of_X := Sum_of_X + data_array[i];
         Sum_of_X2 := Sum_of_X2 + (data_array[i] *
                                   data_array[i]);
      end;
   A := number_of_points * Sum_of_X2 ;
   B := Sum_of_X * Sum_of_X ;
   C := number_of_points * (number_of_points - 1);

   stad_deviation := sqrt((A - B) / C) ;

(* ********************************************** *)

   Write('The standard deviation of the noise is : ');
   Writeln(stad_deviation);

   noise := 1.96 * stad_deviation ;
   i:=0;
   repeat
      begin
         for j:=1 to 8 do
            begin
            if (data_array[i+j]<noise) and
               (data_array[i+j]>(noise*(-1)))
               then  data_array[i+j] := 0 ;
            write(output_file,round(data_array[i+j]):10);
            end;
         writeln(output_file);
         i:=i+8;
      end;
   until i=4096 ;

end.
```

```
program search (input,output);

(* ********************************************************** *)
(*                                                            *)
(*  PROGRAM SEARCH                                 12/29/89   *)
(*                                                            *)
(*   This program performs the reverse search of the          *)
(*   library of the clipped spectra (1,0,-1). At the          *)
(*   beginning it asks for the name of the unknown            *)
(*   spectrum (the spectrum should first be unfolded,         *)
(*   transformed, filtered and clipped) and then it does      *)
(*   the search against all the files in the SEARCH.FIL       *)
(*   file.  The results go into the file                      *)
(*          [.CLIP.SEARCH](unknown).OUT                       *)
(*   where (unknown) is the name of the unknown spectrum.     *)
(*                                                            *)
(*                                  George Asimopoulos        *)
(*                                                            *)
(* ********************************************************** *)

type
   name                  = packed array[1..50] of char ;

   point                 = array[1..4096] of integer ;

var
   i, j, k, positive_points,
   no_points, negative_points              : integer ;
   positive_negative                          : real ;
   unknown_point,reference_point      : point ;
   unknown, reference, results        : name ;
   unknown_file, reference_file,
   input_file, results_file           : text ;

begin
   write('What is the unknown spectrum : ');
   readln(unknown);
   open(unknown_file,unknown,old);
   reset(unknown_file);

   for i:=1 to 4096 do
      read(unknown_file,unknown_point[i]);

   close(unknown_file);

(* Open and prepare the heading for the results file *)

   results := '[.clip.search]';
   i:=7;
```

```
   repeat
      i:=i+1;
      results[i+7] := unknown[i];
   until unknown[i]='.';
   results[i+8]:='o';
   results[i+9]:='u';
   results[i+10]:='t';
   open(results_file,results);
   rewrite(results_file);
   writeln(results_file);
   write(results_file,'   REVERSE SEARCH FOR     ');
   i:=7;
   repeat
      i:=i+1;
      write(results_file,unknown[i]);
   until unknown[i]='.';
   writeln(results_file);
   writeln(results_file);
   write(results_file,'   SPECTRUM              Positive  ');
  writeln(results_file,'NO    Negative  Positive/Negative');
   writeln(results_file);

(* ********************************************************* *)

   open(input_file,'search.fil',old);
   reset(input_file);

   repeat
      readln(input_file,reference);
      open(reference_file,reference,old);
      reset(reference_file);

      for i:=1 to 4096 do
         read(reference_file,reference_point[i]);

      positive_points := 0 ;
      no_points := 0 ;
      negative_points := 0 ;

      for i:=1 to 4096 do
         begin
            if (reference_point[i]=1) and
               (unknown_point[i]=1)
               then  positive_points := positive_points + 1;
            if (reference_point[i]=-1) and
               (unknown_point[i]=-1)
               then  positive_points := positive_points + 1;
            if (reference_point[i]=1) and
               (unknown_point[i]=0)
```

```
                then  no_points := no_points + 1 ;
             if (reference_point[i]=-1) and
                (unknown_point[i]=0)
                then  no_points := no_points + 1 ;
             if (reference_point[i]=1) and
                (unknown_point[i]=-1)
                then negative_points := negative_points + 1 ;
             if (reference_point[i]=-1) and
                (unknown_point[i]=1)
                then negative_points := negative_points + 1 ;
          end;

       if negative_points>0
          then
             positive_negative := positive_points /
                                  negative_points
          else positive_negative := 10 ;
       write(results_file,'   ',reference[8],reference[9]);
       write(results_file,'            ');
       write(results_file,positive_points,no_points,
                negative_points);
       writeln(results_file,'      ',positive_negative:7:4);
       close(reference_file);

   until EOF(input_file);
end.
```

```
C      JUNE 6, 1991
C
C      THIS PROGRAM SOLVES A SYSTEM OF SIMULTANEUS EQUATIONS
C      USING THE METHOD OF LEAST SQUARES AND ALSO IT FORCES
C      THE SOLUTION TO HAVE ONLY NON-NEGATIVE COEFFICIENTS:
C        i.e.    Ax=B AND x>=0
C      AT THE END IT PRINTS THE SIX X COEFFICIENTS, THE
C      EUCLIDIAN NORMAL AND THE MODE WITH THE NNLS SUBROUTING
C      EXITED.
C      THE MAIN SUBROUTINES OF THIS PROGRAM WERE TAKEN FORM
C      LAWSON AND HANSON, "SOLVING LEAST SQUARES PROBLEMS",
C      PRENTICE-HALL, 1974.
C
C      GEORGE ASIMOPOULOS
C
       DIMENSION A(128,6),B(128),X(6),W(6),Z(128),INDEX(6)
       DIMENSION A1(128),A2(128),A3(128)
       DIMENSION A4(128),A5(128),A6(128)
C
C      READ THE MATRIX A FROM THE INPUT FILE
C
       DO 10 I=1,128
    10 READ(3,100) A1(I),A2(I),A3(I),A4(I),A5(I),A6(I),B(I)
       DO 20 I=1,128
       A(I,1)=A1(I)
       A(I,2)=A2(I)
       A(I,3)=A3(I)
       A(I,4)=A4(I)
       A(I,5)=A5(I)
       A(I,6)=A6(I)
    20 CONTINUE
C
C      CALL THE SUBROUTING NNLS TO FIND THE SOLUTION
C
       CALL NNLS(A,128,128,6,B,X,RNORM,W,Z,INDEX,MODE)
C
C      PRINT OUT THE SOLUTION
C
       DO 30 I=1,6
    30 WRITE(4,105) I,X(I)
       WRITE(4,*)
       WRITE(4,110) RNORM,MODE
       WRITE(4,120)
   100 FORMAT(7F10.0)
   110 FORMAT(3X,'RNORM = ',F10.0,'   MODE = ',I3)
   105 FORMAT(5X,I2,')',2X,F8.4)
   120 FORMAT('*****************************')
       STOP
       END
```

```
      FUNCTION DIFF(X,Y)
      DIFF=X-Y
      RETURN
      END
      SUBROUTINE G1 (A,B,COS,SIN,SIG)
C
C     COMPUTE ORTHOGONAL ROTATION MATRIX
C     COMPUTE..MATRIX (C, S) SO THAT
C          (C, S) (A)=(SQRT(A**2+B**2))
C                    (-S,C)          (-S,C) (B) = (  0  )
C     COMPUTE SIG =SQRT(A**2+B**2)
C         SIG IS COMPUTED LAST TO ALLOW FOR THE POSSIBILITY
C         THAT SIG MAY BE IN THE SAME LOCATION AS A OR B.
C
      ZERO=0.
      ONE=1.
      IF (ABS(A).LE.ABS(B)) GO TO 10
      XR=B/A
      YR=SQRT(ONE+XR**2)
      COS=SIGN(ONE/YR,A)
      SIN=COS*XR
      SIG=ABS(A)*YR
      RETURN
   10 IF (B) 20,30,20
   20 XR=A/B
      YR=SQRT(ONE+XR**2)
      SIN=SIGN(ONE/YR,B)
      COS=SIN*XR
      SIG=ABS(B)*YR
      RETURN
   30 SIG=ZERO
      COS=ZERO
      SIN=ONE
      RETURN
      END
      SUBROUTINE G2    (COS,SIN,X,Y)
C
C          APPLY THE ROTATION COMPUTED BY G1 TO (X,Y).
C
      XR=COS*X+SIN*Y
      Y=-SIN*X+COS*Y
      X=XR
      RETURN
      END
      SUBROUTINE H12
     +(MODE,LPIVOT,L1,M,U,IUE,UP,C,ICE,ICV,NCV)
C
C     CONSTRUCTION AND/OR APPLICATION OF A SINGLE
C     HOUSEHOLDER TRANSFORMATION..   Q=I + U*(U**T)/B
```

```
C
C      MODE    = 1 OR 2   TO SELECT ALGORITHM  H1  OR  H2 .
C      LPIVOT IS THE INDEX OF THE PIVOT ELEMENT.
C      L1,M    IF L1 .LE. M   THE TRANSFORMATION WILL BE
C              CONSTRUCTED TO ZERO ELEMENTS INDEXED FROM
C              L1 THROUGH M.       IF L1 GT. M
C              THE SUBROUTINE DOES AN IDENTITY TRANSFORMATION.
C      U(),IUE,UP     ON ENTRY TO H1 U() CONTAINS THE PIVOT
C                     VECTOR. IUE IS THE STORAGE INCREMENT
C                     BETWEEN ELEMENTS.   ON EXIT FROM H1 U()
C                     AND UP CONTAIN QUANTITIES DEFINING THE
C                     VECTOR U OF THE HOUSEHOLDER
C                     TRANSFORMATION.   ON ENTRY TO
C                     H2 U() AND UP SHOULD CONTAIN QUANTITIES
C                     PREVIOUSLY COMPUTED BY H1.  THESE WILL
C                     NOT BE MODIFIED BY H2.
C      C()     ON ENTRY TO H1 OR H2 C() CONTAINS A MATRIX
C              WHICH WILL BE REGARDED AS A SET OF VECTORS TO
C              WHICH THE HOUSEHOLDER TRANSFORMATION IS TO BE
C              APPLIED.  ON EXIT C() CONTAINS THE SET OF
C              TRANSFORMED VECTORS.
C      ICE     STORAGE INCREMENT BETWEEN ELEMENTS OF VECTORS
C              IN C()
C      ICV     STORAGE INCREMENT BETWEEN VECTORS IN C().
C      NCV     NUMBER OF VECTORS IN C() TO BE TRANSFORMED. IF
C              NCV .LE. 0 NO OPERATIONS WILL BE DONE ON C().
C
      DIMENSION U(IUE,M),C(1)
      DOUBLE PRECISION SM,B
      ONE=1.
C
      IF (0.GE.LPIVOT.OR.LPIVOT.GE.L1.OR.L1.GT.M) RETURN
      CL=ABS(U(1,LPIVOT))
      IF (MODE.EQ.2) GO TO 60
C
C                     *******CONSTRUCT THE TRANSFORMATION ***
C
      DO 10 J=L1,M
   10      CL=AMAX1(ABS(U(1,J)),CL)
      IF (CL) 130,130,20
   20 CLINV=ONE/CL
      SM=(DBLE(U(1,LPIVOT))*CLINV)**2
           DO 30 J=L1,M
   30      SM=SM+(DBLE(U(1,J))*CLINV)**2
C
C               CONVERT DBLE. PREC. SM TO SINGLE PREC. SM1
C
      SM1=SM
      CL=CL*SQRT(SM1)
```

```
      IF (U(1,LPIVOT)) 50,50,40
   40 CL=-CL
   50 UP=U(1,LPIVOT)-CL
      U(1,LPIVOT)=CL
      GO TO 70
C
C     *****APPLY THE TRANSFORMATION I+U*(U**T)/B TO C ******
C
   60 IF (CL) 130,130,70
   70 IF (NCV.LE.0) RETURN
      B=DBLE(UP)*U(1,LPIVOT)
C
C               B MUST BE NONPOSITIVE HERE. IF B=0. , RETURN
C
      IF (B) 80,130,130
   80 B=ONE/B
      I2=1-ICV+ICE*(LPIVOT-1)
      INCR=ICE*(L1-LPIVOT)
           DO 120 J=1, NCV
           I2=I2+ICV
           I3=I2+INCR
           I4=I3
           SM=C(I2)*DBLE(UP)
               DO 90 I=L1,M
               SM=SM+C(I3)*DBLE(U(1,I))
   90          I3=I3+ICE
           IF (SM) 100,120,100
  100      SM=SM*B
           C(I2)=C(I2)+SM*DBLE(UP)
               DO 110 I=L1,M
               C(I4)=C(I4)+SM*DBLE(U(1,I))
  110          I4=I4+ICE
  120      CONTINUE
  130 RETURN
      END
C
      SUBROUTINE NNLS (A,MDA,M,N,B,X,RNORM,W,ZZ,INDEX,MODE)
C
C               *******NONNEGATIVE LEAST SQUARES *****
C
C     GIVEN AN M BY N MATRIX, A, AND AN M-VECTOR, B, COMPUTE
C     AN N-VECTOR, X, WHICH SOLVES THE LEAST SQUARES PROBLEM
C
C                        A*X=B SUBJECT TO X .GE.0
C
C     A(),MDA,M,N      MDA IS THE FIRST DIMENSIONING
C                      PARAMETER FOR THE  ARRAY,A(). ON ENTRY
C                      A() CONTAINS THE M BY N MATRIX, A. ON
C                      EXIT A() CONTAINS THE PRODUCT MATRIX,
```

```
C                       Q*A. WHERE Q IS ANC M BY M ORTHOGONAL
C                       MATRIX GENERATED IMPLICITLY BY
C                       THIS SUBROUTINE.
C     B()       ON ENTRY B() CONTAINS THE M VECTOR,B.  ON EXIT
C               B() CONTAINS Q*B.
C     X()       ON ENTRY X() NEED NOT BE INITIALIZED. ON EXIT
C               X() WILL CONTAIN THE SOLUTION VECTOR.
C     RNORM     ON EXIT RNORM  CONTAINS THE EUCLIDEAN NORM OF
C               THE RESIDUAL VECTOR.
C     W()       AN N-ARRAY OF WORKING SPACE, ON EXIT W()  WILL
C               CONTAIN THE DUAL SOLUTION VECTOR. W WILL
C               SATISFY W(I)=0. FOR ALL I IN SET P AND W(I)
C               .LE.0. FOR ALL I IN SET Z
C     ZZ()      AN M-ARRAY OF WORKING SPACE.
C     INDEX()   AN INTEGER WORKING SPACE OR ARRAY OF LENGTH AT
C               LEAST N ON EXIT THE CONTENTS OF THIS ARRAY
C               DEFINE THE SETS P AND Z AS FOLLOWS..
C
C                   INDEX(1)  THRU INDEX(NSTEP) = SET P.
C                   INDEX(IZ1) THRU INDEX (IZ2)   = SET Z.
C                   IZ1 = NSTEP+1 =NPP1
C                   IZ2 = N
C     MODE       THIS IS A SUCCESS-FAILURE FLAG WITH THE
C                FOLLOWING MEANINGS.
C                1.     THE SOLUTION HAS BEEN COMPUTED
C                       SUCCESSFULLY
C                2.     THE DIMENSIONS OF THE PROBLEM ARE BAD.
C                       EITHER M.LE.0 OR N.LE.0.
C                3.     ITERATION COUNT EXCEEDED. MORE THAN 3*N
C                       ITERATIONS.
C
      DIMENSION A(MDA,N), B(M), X(N), W(N), ZZ(M)
      INTEGER INDEX(N)
      ZERO=0.
      ONE=1.
      TWO=2.
      FACTOR=0.01
C
      MODE=1
      IF (M.GT.0.AND.N.GT.0) GO TO 10
      MODE=2
      RETURN
   10 ITER=0
      ITMAX=3*N
C
C                  INITIALIZE THE ARRAYS INDEX() AND X().
C
           DO 20 I=1,N
           X(I)=ZERO
```

```
   20      INDEX(I)=I
C
      IZ2=N
      IZ1=1
      NSETP=0
      NPP1=1
      ITERA=0
C
C                    ******* MAIN LOOP BEGINS HERE *******
C
   30 CONTINUE
      ITERA=ITERA + 1
      PRINT *,' LOOP A ITERATION : ',ITERA
      ITERB=0
C
C              QUIT IF ALL COEFFICIENTS ARE ALREADY IN THE
C              SOLUTION.
C              OR IF M COLS OF A HAVE BEEN TRIANGULARIZED.
C
      IF (IZ1.GT.IZ2.OR.NSETP.GE.M) GO TO 350
C
C           COMPUTE COMPONENTS OF THE DUAL (NEGATIVE
C                                GRADIENT) VECTOR W()
C
           DO 50 IZ=IZ1,IZ2
           J=INDEX(IZ)
           SM=ZERO
               DO 40 L=NPP1,M
   40          SM=SM+A(L,J)*B(L)
   50      W(J)=SM
C
C                          FIND LARGEST POSITIVE W(J).
C
   60 WMAX=ZERO
           DO 70 IZ=IZ1,IZ2
           J=INDEX(IZ)
           IF (W(J).LE.WMAX) GO TO 70
           WMAX=W(J)
           IZMAX=IZ
   70      CONTINUE
C
C                IF WMAX.LE.0 GO TO TERMINATION.
C                THIS INDICATES SATISFACTION OF THE KUHN
C                TUCKER CONDITIONS.
C
      IF (WMAX) 350,350,80
   80 IZ=IZMAX
      J=INDEX(IZ)
C
```

```
C     THE SIGN OF W(J) IS OK FOR J TO BE MOVED TO SET P.
C     BEGIN THE TRANSFORMATION AND CHECK NEW DIAGONAL
C     ELEMENT TO AVOID NEAR LINEAR DEPENDENCE.
C
      ASAVE=A(NPP1,J)
      CALL H12 (1,NPP1,NPP1+1,M,A(1,J),1,UP,DUMMY,1,1,0)
      UNORM=ZERO
      IF (NSETP.EQ.0) GO TO 100
           DO 90 L=1,NSETP
   90      UNORM=UNORM+A(L,J)**2
  100 UNORM=SQRT(UNORM)
      IF (DIFF(UNORM+ABS(A(NPP1,J))*FACTOR,UNORM))
     +130,130,110
C
C     COL J IS SUFFICIENTLY INDEPENDENT. COPY B INTO ZZ,
C     UPDATE ZZ AND
C     SOLVE FOR ZTEST ( = PROPOSED NEW VALUE FOR X(J) ).
C
  110      DO 120 L=1,M
  120      ZZ(L)=B(L)
      CALL H12 (2,NPP1,NPP1+1,M,A(1,J),1,UP,ZZ,1,1,1)
      ZTEST=ZZ(NPP1)/A(NPP1,J)
C
C                                     SEE IF ZTEST IS POSITIVE
C
      IF (ZTEST) 130,130,140
C
C     REJECT J AS A CANDIDATE TO BE MOVED FROM SET Z TO SET
C     P.  RESTORE A(NPP1,J), SET W(J)=0., AND LOOP BACK TO
C     TEST DUAL COEFFS AGAIN.
C
  130 A(NPP1,J)=ASAVE
      W(J)=ZERO
      GO TO 60
C
C     THE INDEX J=INDEX(IZ) HAS BEEN SELECTED TO BE MOVED
C     FROM SET Z TO SET P.   UPDATE B,  UPDATE INDICES, -
C     APPLY HOUSEHOLDER TRANSFORMATIONS TO COLS IN NEW SET
C     Z, ZERO SUBDIAGONAL  ELTS IN COL J, SET W(J)=0.
C
  140      DO 150 L=1,M
  150      B(L)=ZZ(L)
C
      INDEX(IZ)=INDEX(IZ1)
      INDEX(IZ1)=J
      IZ1=IZ1+1
      NSETP=NPP1
      NPP1=NPP1+1
C
```

```
      IF (IZ1.GT.IZ2) GO TO 170
           DO 160 JZ=IZ1,IZ2
           JJ=INDEX(JZ)
  160      CALL H12
     +(2,NSETP,NPP1,M,A(1,J),1,UP,A(1,JJ),1,MDA,1)
  170 CONTINUE
C
      IF (NSETP.EQ.M) GO TO 190
           DO 180 L=NPP1,M
  180      A(L,J)=ZERO
  190 CONTINUE
C
      W(J)=ZERO
C
C                    SOLVE THE TRIANGULAR SYSTEM.
C                    STORE THE SOLUTION TEMPORARILY IN ZZ().
C
      ASSIGN 200 TO NEXT
      GO TO 400
  200 CONTINUE
      PRINT *,'   LOOP B ITERATION'
C
C                     *******SECONDARY LOOP BEGINS HERE ***
C
C                                ITERATION COUNTER.
C
  210 ITER=ITER+1
      ITERB=ITERB + 1
      PRINT *,'                      ',ITERB
      IF (ITER.LE.ITMAX) GO TO 220
      MODE=3
      WRITE (6,440)
      GO TO 350
  220 CONTINUE
C
C              SEE IF ALL NEW CONSTRAINED COEFFS ARE FEASIBLE
C                                  IF NOT COMPUTE ALPHA -
C
      ALPHA=TWO
           DO 240 IP=1,NSETP
           L=INDEX(IP)
           IF (ZZ(IP)) 230,230,240
C
  230      T=-X(L)/(ZZ(IP)-X(L))
           IF (ALPHA.LE.T) GO TO 240
           ALPHA=T
           JJ=IP
  240      CONTINUE
C
```

```
C           IF ALL NEW CONSTRAINED COEFFS ARE FEASIBLE THEN
C           ALPHA WILL STILL=2.      IF SO EXIT FROM SECONDARY
C           LOOP TO MAIN LOOP.
C
      IF (ALPHA.EQ.TWO) GO TO 330
C
C           OTHERWISE USE ALPHA WHICH WILL BE BETWEEN
C           0. AND 1.   TO INTERPOLATE BETWEEN THE OLD X AND
C           THE NEW ZZ.
C
            DO 250 IP=1,NSETP
            L=INDEX(IP)
  250       X(L)=X(L)+ALPHA*(ZZ(IP)-X(L))
C
C        MODIFY A AND B AND THE INDEX  ARRAYS TO MOVE
C        COEFFICIENT I FROM SET P TO SET Z.
C
      I=INDEX(JJ)
  260 X(I)=ZERO
C
      IF (JJ.EQ.NSETP) GO TO 290
      JJ=JJ+1
            DO 280 J=JJ,NSETP
            II=INDEX(J)
            INDEX(J-1)=II
            CALL G1 (A(J-1,II),A(J,II),CC,SS,A(J-1,II))
            A(J,II)=ZERO
                  DO 270 L=1,N
                  IF (L.NE.II) CALL G2(CC,SS,A(J-1,L),A(J,L))
  270             CONTINUE
  280       CALL G2 (CC,SS,B(J-1),B(J))
  290 NPP1=NSETP
      NSETP=NSETP-1
      IZ1=IZ1-1
      INDEX(IZ1)=I
C
C        SEE IF THE REMAINING COEFFS IN SET P ARE FEASIBLE.
C        THEY SHOULD BE BECAUSE OF THE WAY ALPHA WAS
C        DETERNINED. IF ANY ARE INFEASIBLE IT IS DUE TO
C        ROUND-OFF ERROR. ANY THAT ARE NONPOSITIVE WILL BE
C        SET TO ZERO  AND MOVED FROM SET P TO SET Z.
C
            DO 300 JJ=1,NSETP
            I=INDEX(JJ)
            IF (X(I)) 260,260,300
  300       CONTINUE
C
C     COPY B( ) INTO ZZ( ). THEN SOLVE AGAIN AND LOOP BACK
C
```

```
            DO 310 I=1,M
  310       ZZ(I)=B(I)
      ASSIGN 320 TO NEXT
      GO TO 400
  320 CONTINUE
      GO TO 210
C
C                        ***** END OF SECONDARY LOOP**********
C
  330       DO 340 IP=1,NSETP
            I=INDEX(IP)
  340       X(I)=ZZ(IP)
C
C         ALL NEW COEFFS ARE POSITIVE. LOOP BACK TO BEGINNING
C
      GO TO 30
C
C                          *******END OF MAIN LOOP ***********
C
C              COME TO HERE FOR TERMINATION.
C              COMPUTE THE NORM OF THE FINAL RESIDUAL VECTOR.
C
  350 SM=ZERO
      IF (NPP1.GT.M) GO TO 370
            DO 360 I=NPP1,M
  360       SM=SM+B(I)**2
      GO TO 390
  370       DO 380 J=1,N
  380       W(J)=ZERO
  390 RNORM=SQRT(SM)
      RETURN
C
C     THE FOLLOWING BLOCK OF CODE IS USED AS AN INTERNAL
C     SUBROUTINE TO SOLVE THE TRIANGULAR SYSTEM, PUTTING THE
C     SOLUTION IN ZZ().
  400       DO 430 L=1,NSETP
            IP=NSETP+1-L
            IF (L.EQ.1) GO TO 420
                DO 410 II=1,IP
  410           ZZ(II)=ZZ(II)-A(II,JJ)*ZZ(IP+1)
  420       JJ=INDEX(IP)
  430       ZZ(IP)=ZZ(IP)/A(IP,JJ)
      GO TO NEXT, (200,320)
  440 FORMAT (35H0 NNLS QUITTING ON ITERATION COUNT.)
      END
```

Appendix B.
Composition of the 40 unknowns.

One component unknowns - Unknown 1 to 10.

Unknown #	Component		Factor
1	AN	Anthracene	1
2	BB	BBOT	1
3	BD	BBD	1
4	FL	Fluorene	1
5	MA	9-Methylanthracene	1
6	MN	2-Methylnaphthalene	1
7	PD	PPD	1
8	PO	POPOP	1
9	PP	PPO	1
10	TQ	Triphenylamine(1.00e-5)	1

Two component unknowns - Mixture 1 to 10.

Mixture #	Component		Factor
1	IN	1-Naphthol	3
	BB	BBOT	6
2	CA	9,10-Dichloroanthracene	1
	PY	Pyrene	5
3	PY	Pyrene	1
	AR	Acridine	2
4	AN	Anthracene	3
	IB	1,1-Binaphthyl	9
5	IN	1-Naphthol	3
	TM	Triphenylamine(1.00e-3)	8
6	AZ	Azulene	5
	VA	9-Vinylanthracene	2
7	CA	9,10-Dichloroanthracene	1
	ID	Indole	3
8	BB	BBOT	4
	AQ	Anthraquinone	1
9	NP	aNPO	1
	QP	p-Quaterphenyl	2
10	QP	p-Quaterphenyl	7
	QU	Quinoline	2

Three component unknowns - Mixture 11 to 20.

Mixture #	Component		Factor
11	PN	2-Phenylnaphthalene	8
	EA	Methylanthracene	10
	DA	9,10-Diphenylanthracene	9
12	IB	1,1-Binaphthyl	2
	TE	Tetracene	7
	PD	PPD	3
13	FL	Fluorene	3
	BO	BBO	3
	DI	4,5-Diphenylimidazule	6
14	TA	Triphenylamine	5
	PA	9-Phenylanthracene	6
	TN	Triphenylamine(5.00e-4)	5
15	IM	1-Methylnaphthalene	8
	PE	Perylene	9
	TQ	Triphenylamine(1.00e-5)	8
16	PB	PBD	9
	DA	9,10-Diphenylanthracene	1
	QU	Quinoline	2
17	CA	9,10-Dichloroanthracene	10
	BE	4-Biphenylphenylether	4
	PH	Phenanthrene	5
18	IA	1-Aminoanthracene	1
	VB	4-Vinylbiphenyl	1
	BI	2,2-Binaphthyl	10
19	PE	Perylene	3
	CA	9,10-Dichloroanthracene	7
	PH	Phenanthrene	3
20	TE	Tetracene	1
	BD	BBD	4
	TS	Triphenylamine(1.00e-4)	10

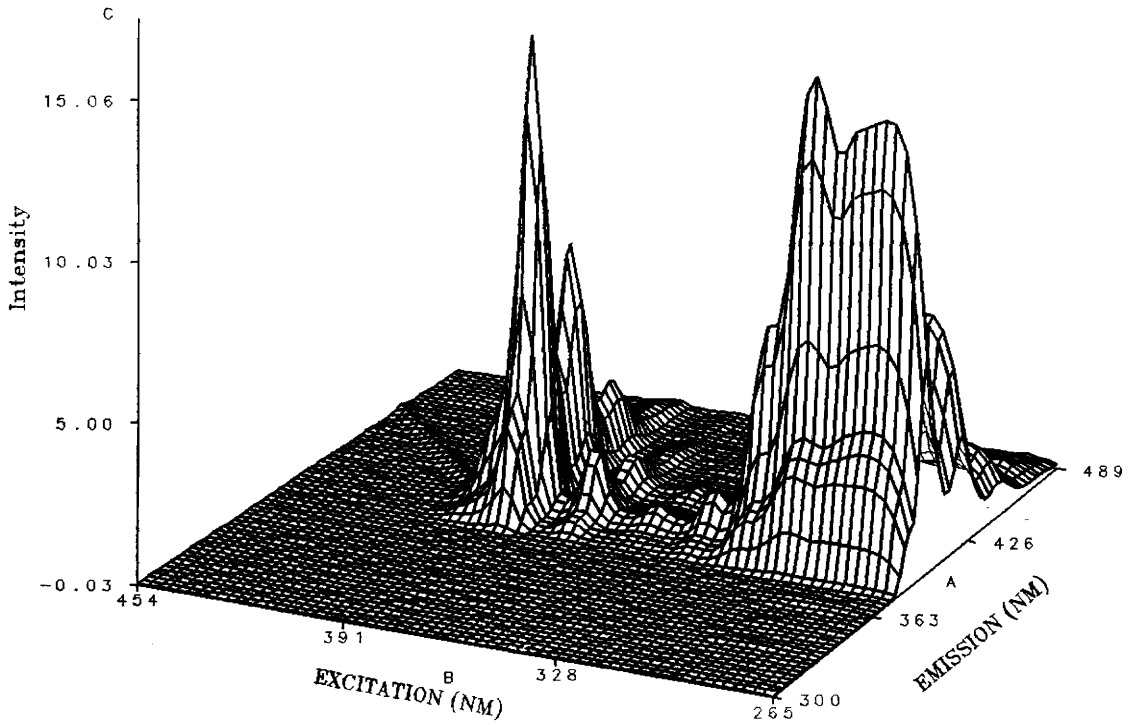
Five component unknowns - Mixture 21 to 30

Mixture #	Component		Factor
21	QP	p-Quaterphenyl	8
	NP	aNPO	2
	IA	1-Aminoanthracene	7
	PP	PPO	7
	AN	Anthracene	5
22	MN	2-Methylnaphthalene	8
	PE	Perylene	6
	PD	PPD	3
	BD	BBD	3
	TE	Tetracene	3
23	MA	9-Methylanthracene	2
	TT	Triphenylamine(5.00e-5)	7
	IB	1,1-Binaphthyl	2
	SA	Salicylic Acid	4
	IN	1-Naphthol	5
24	EA	Methylanthracene	3
	TE	Tetracene	2
	AN	Anthracene	4
	ID	Indole	7
	AA	2-Aminoanthracene	6
25	TS	Triphenylamine(1.00e-4)	3
	DP	DimethylPOPOP	9
	DN	2,3-Dimethylnaphthalene	7
	BE	4-Biphenylphenylether	8
	AN	Anthracene	4
26	QP	p-Quaterphenyl	6
	BD	BBD	3
	IP	1-Phenylnaphthalene	6
	BI	2,2-Binaphthyl	4
	PD	PPD	9
27	SA	Salicylic Acid	5
	BI	2,2-Binaphthyl	3
	DA	9,10-Diphenylanthracene	1
	AQ	Anthraquinone	1
	BE	4-Biphenylphenylether	2

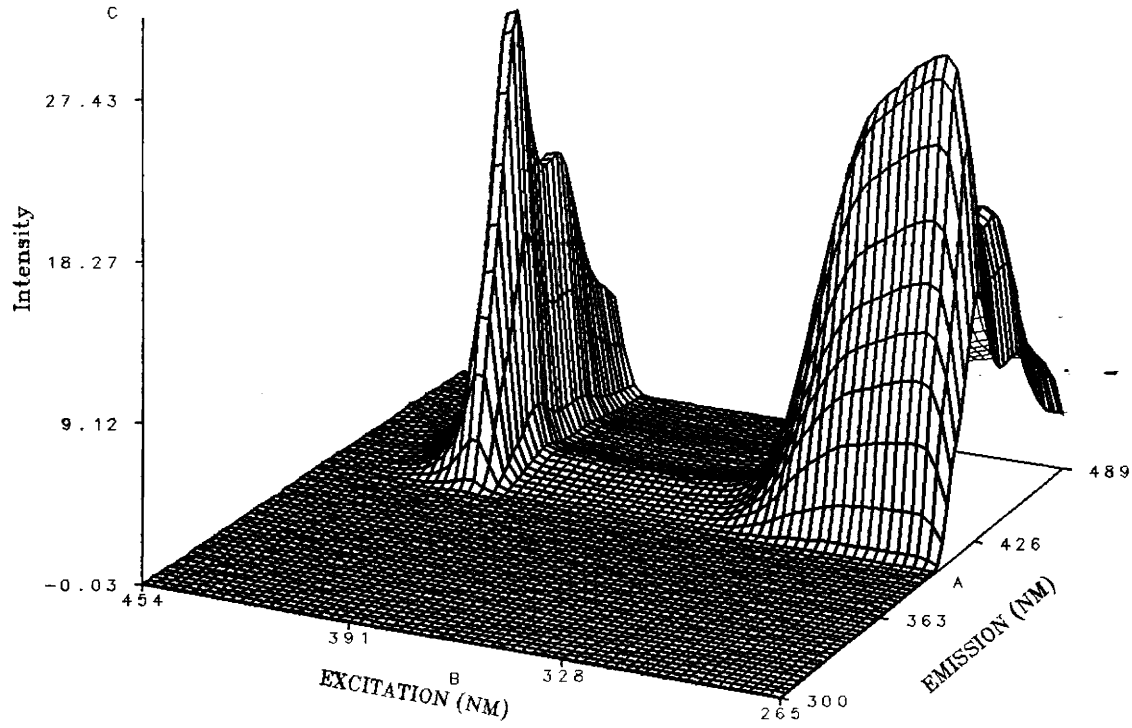
Five component unknowns - Mixture 21 to 30. (continue)

28	TR	Triphenylene	3
	PY	Pyrene	6
	DP	DimethylPOPOP	2
	IN	1-Naphthol	5
	ID	Indole	9
29	AA	2-Aminoanthracene	7
	AC	Anthranilic Acid	1
	TM	Triphenylamine(1.00e-3)	3
	IB	1,1-Binaphthyl	5
	AZ	Azulene	1
30	TR	Triphenylene	9
	IA	1-Aminoanthracene	5
	AA	2-Aminoanthracene	8
	IM	1-Methylnaphthalene	2
	CH	Chrysene	3

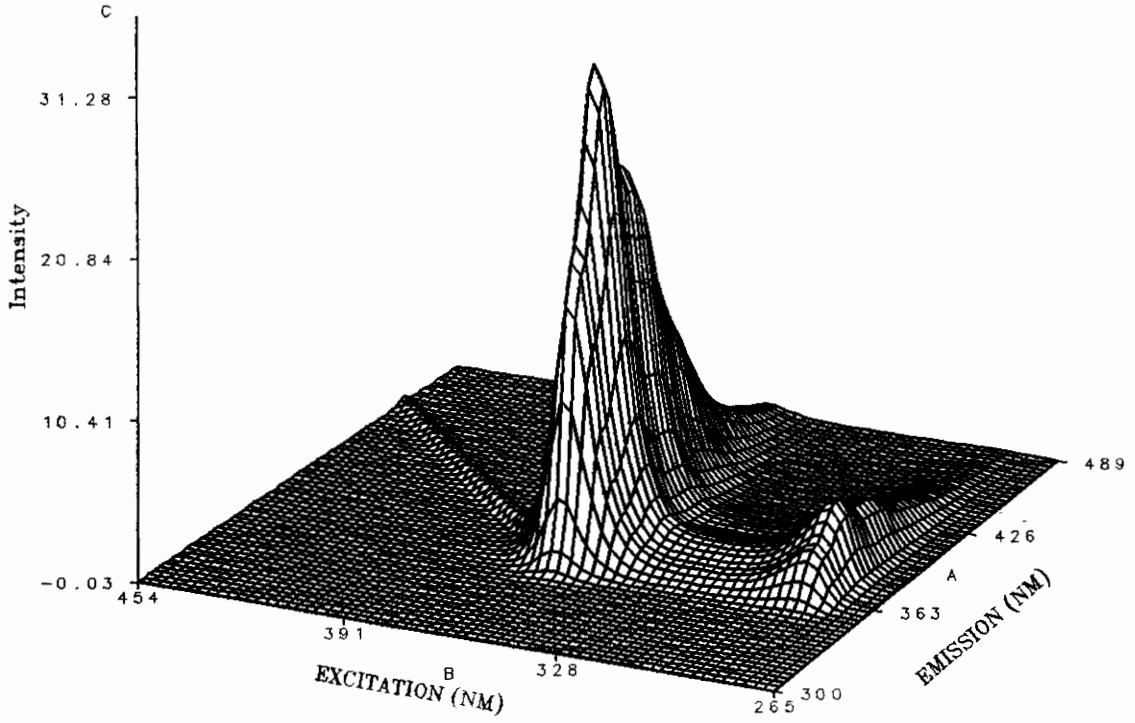
Appendix C.
Excitation Emission Matrices of 40 unknowns.



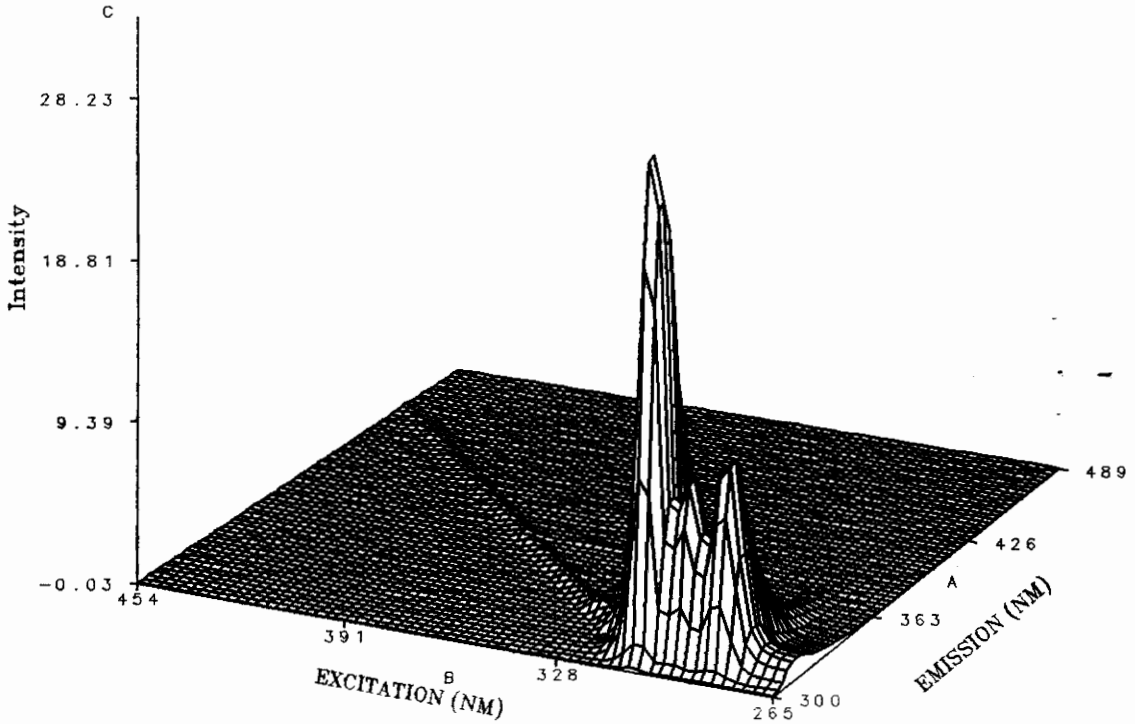
Unknown 1. (one component)



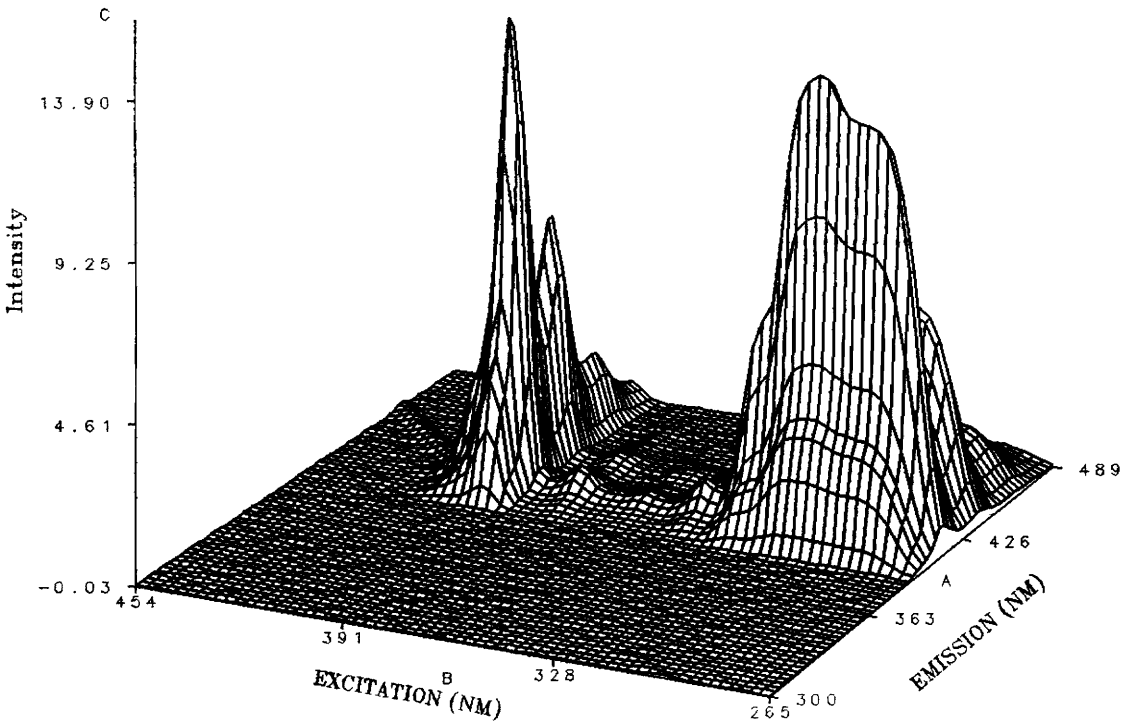
Unknown 2. (one component)



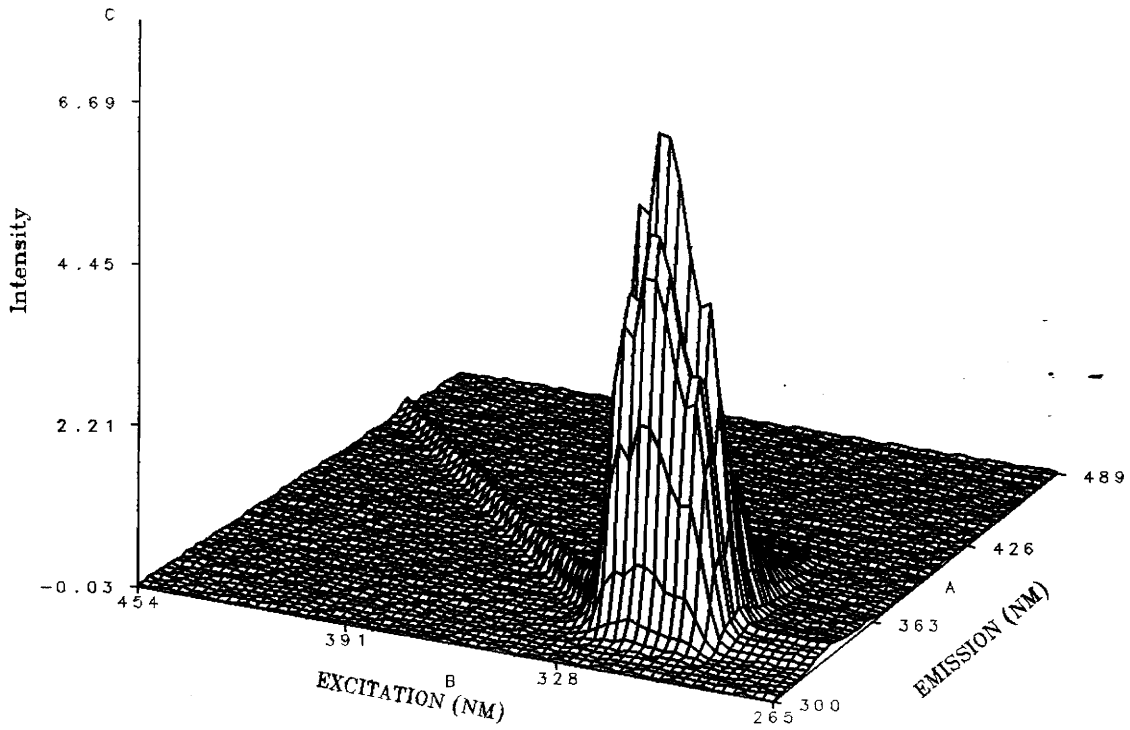
Unknown 3. (one component)



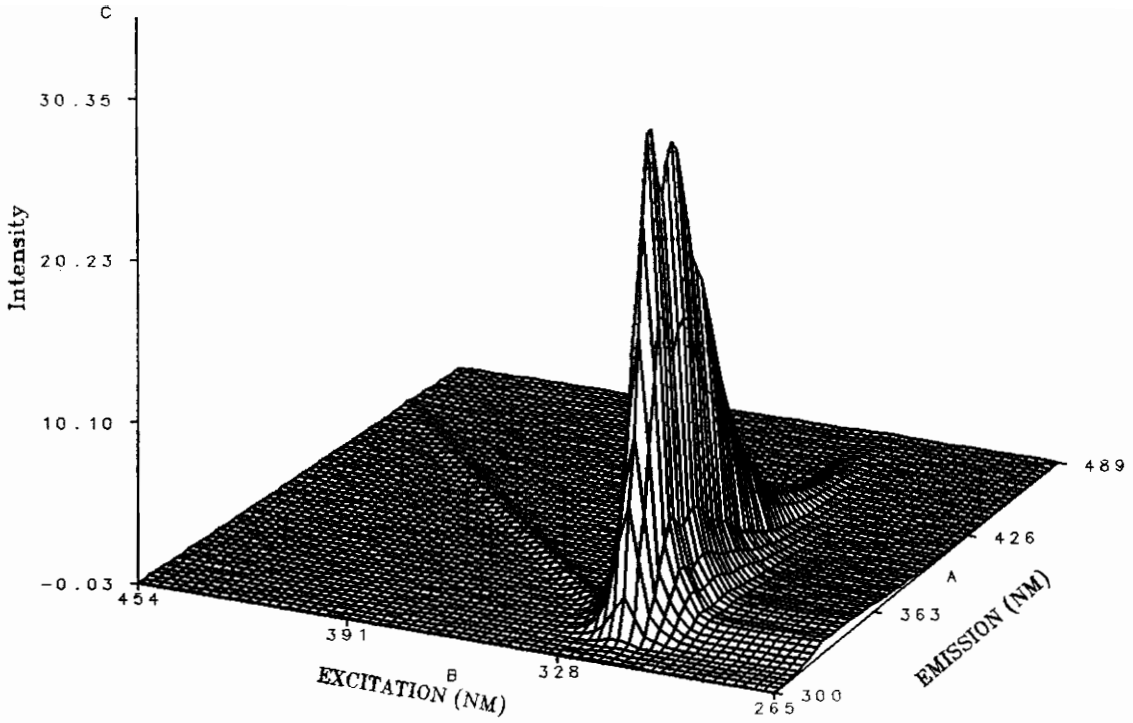
Unknown 4. (one component)



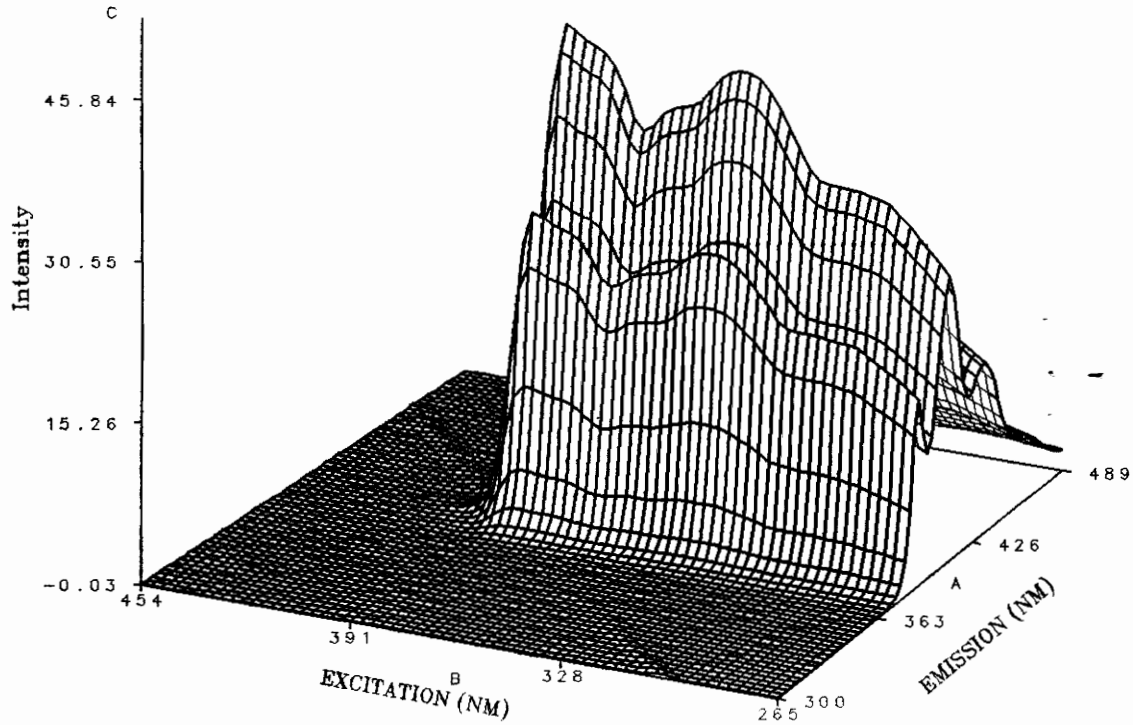
Unknown 5. (one component)



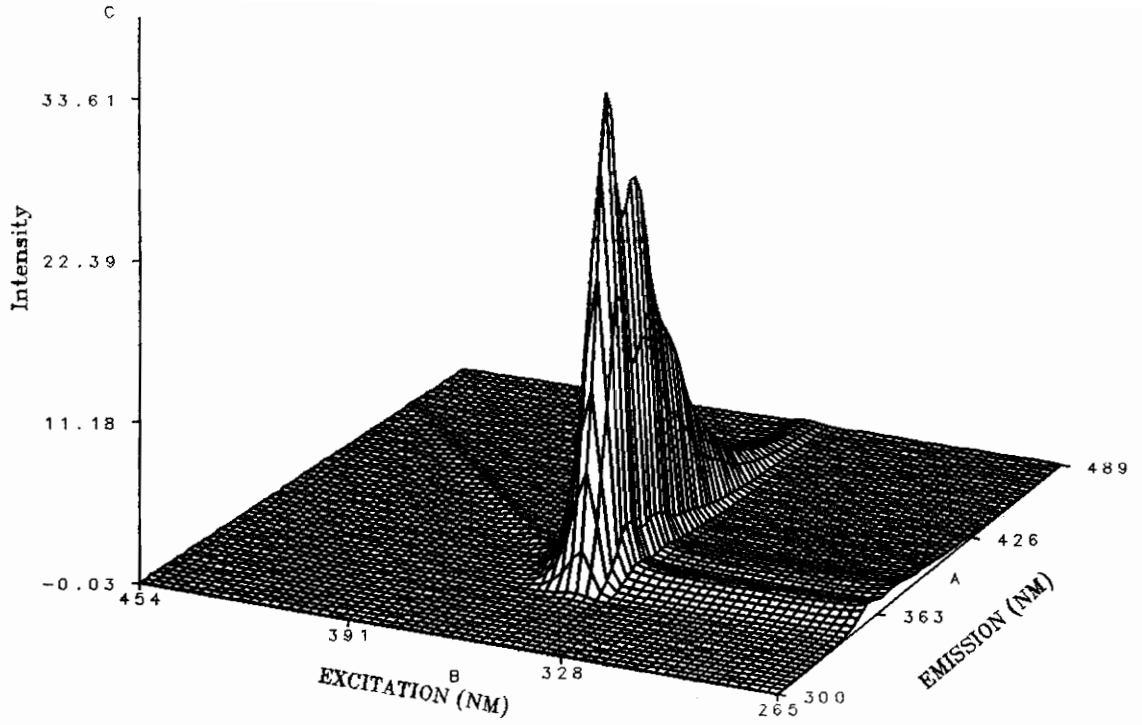
Unknown 6. (one component)



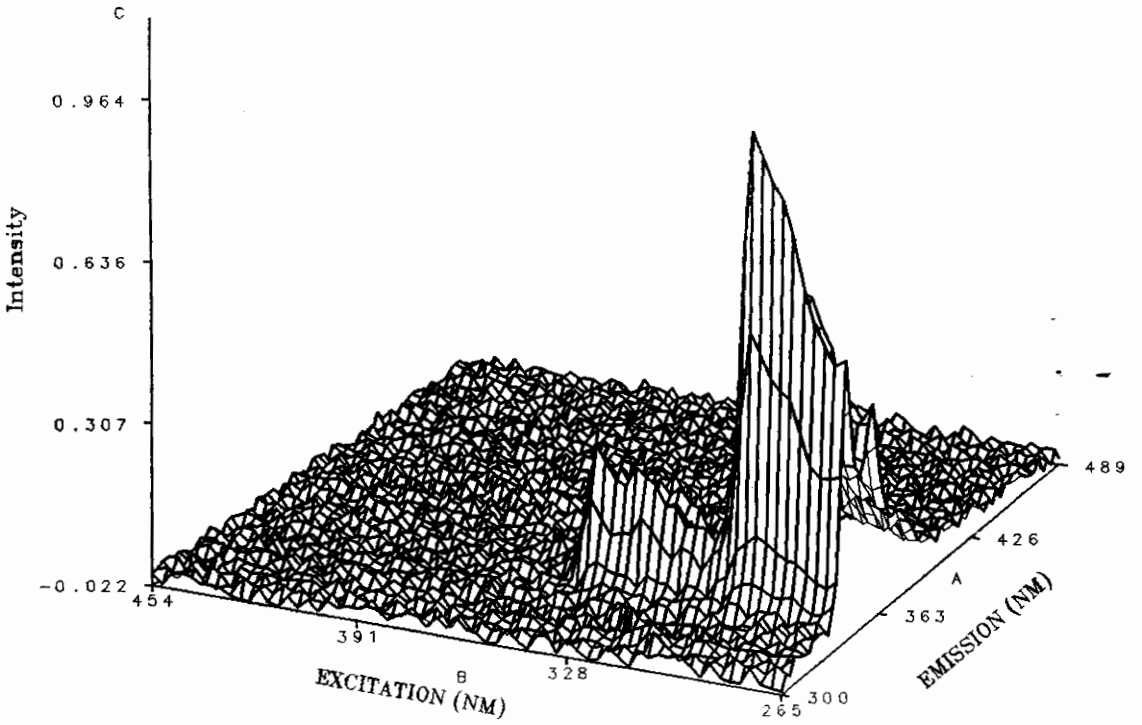
Unknown 7. (one component)



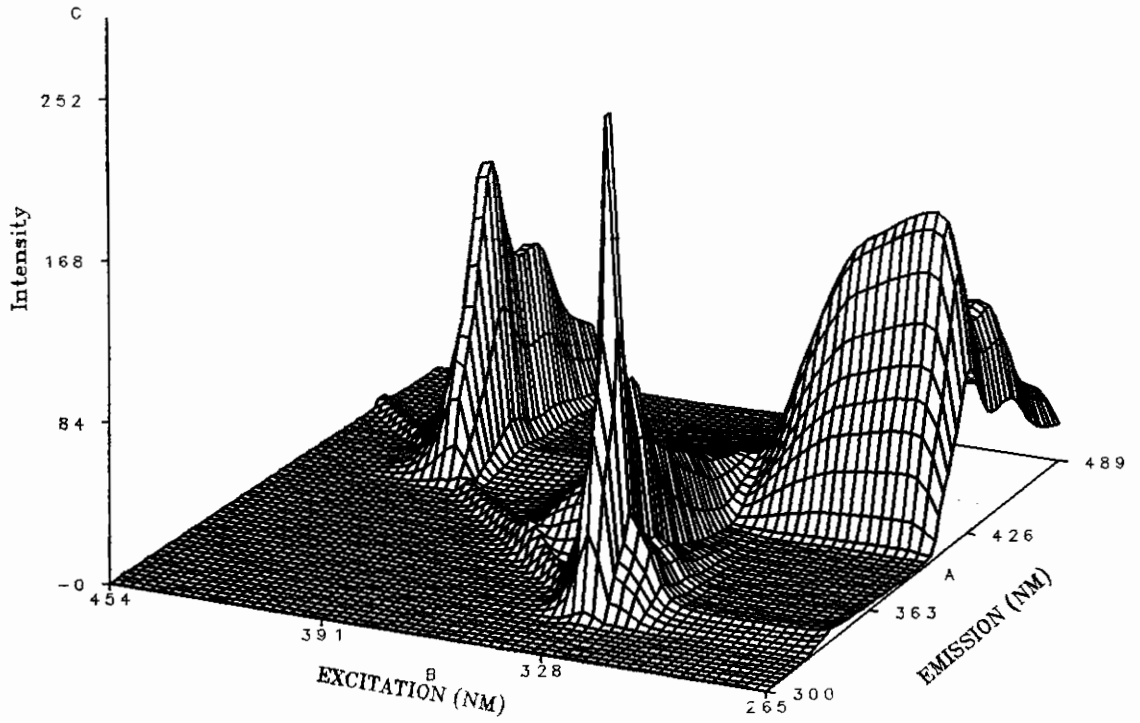
Unknown 8. (one component)



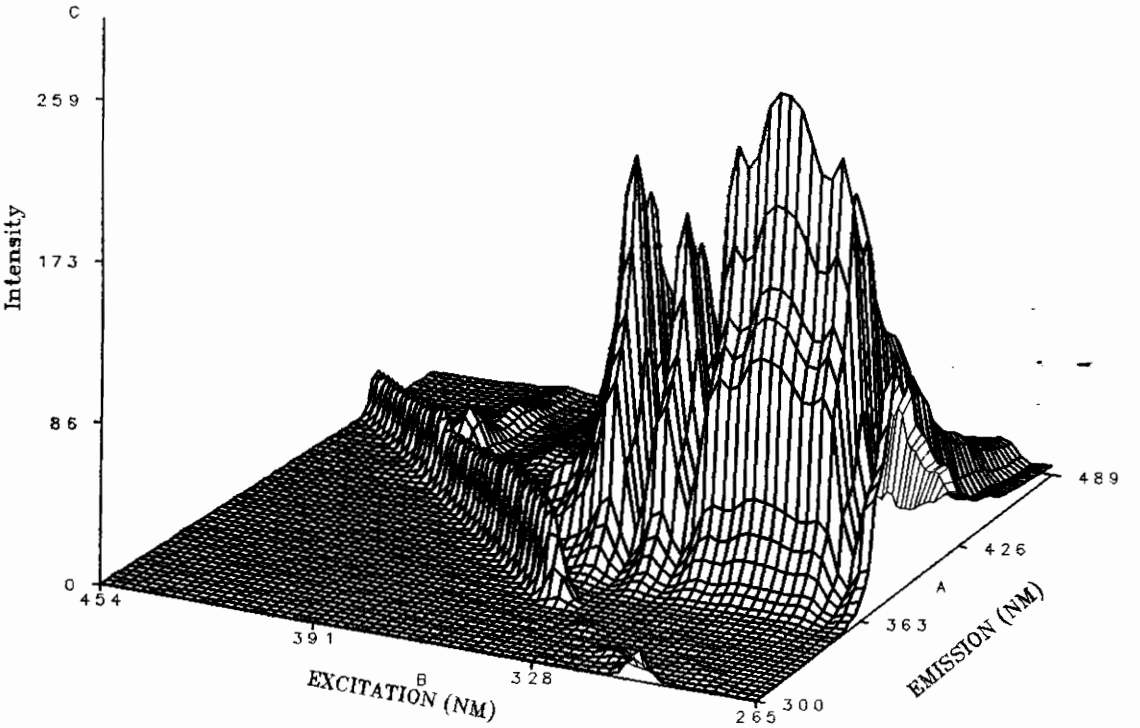
Unknown 9. (one component)



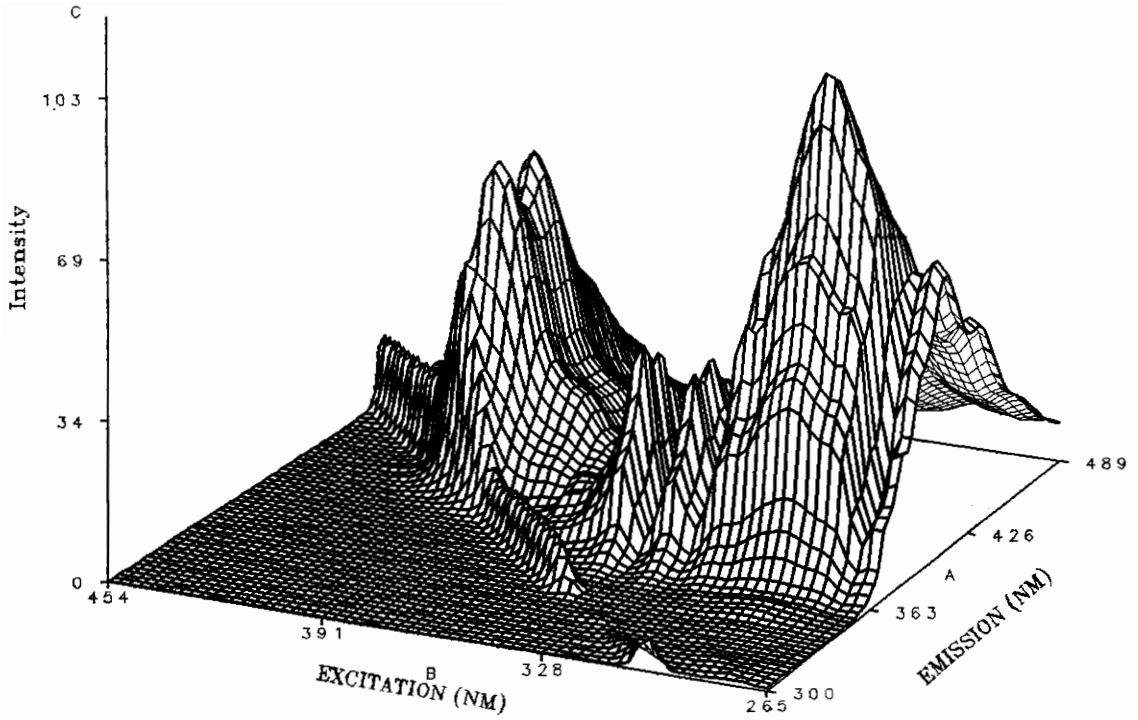
Unknown 10. (one component)



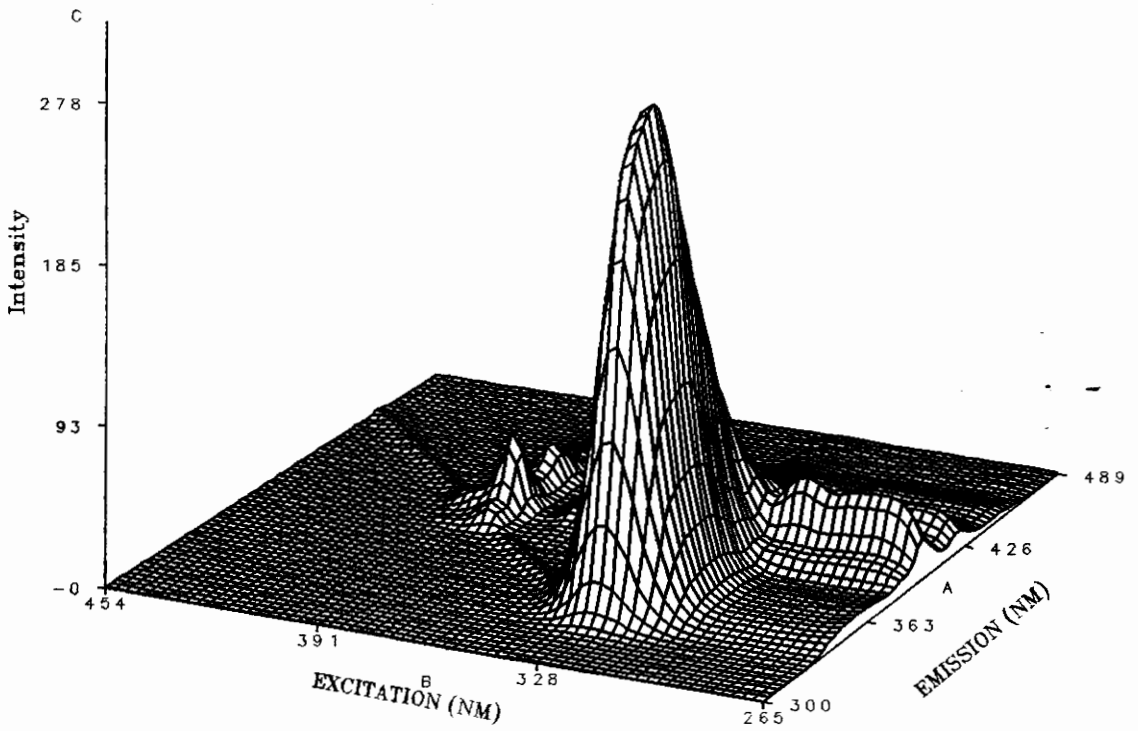
Mixture 1. (two components)



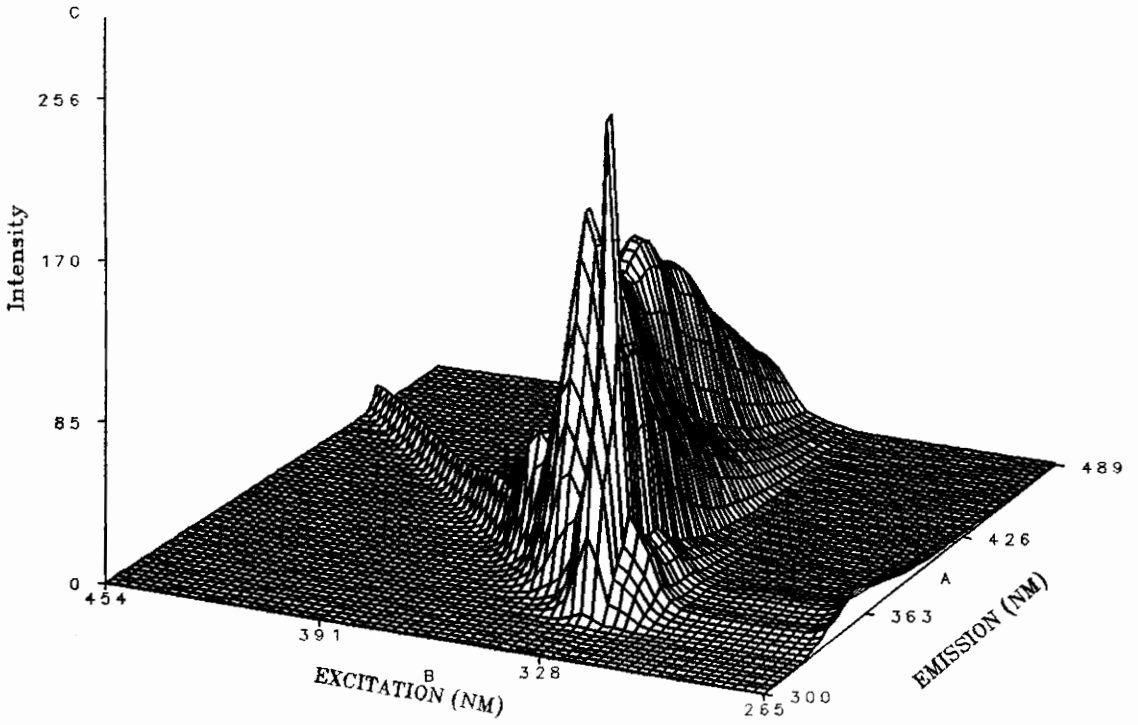
Mixture 2. (two components)



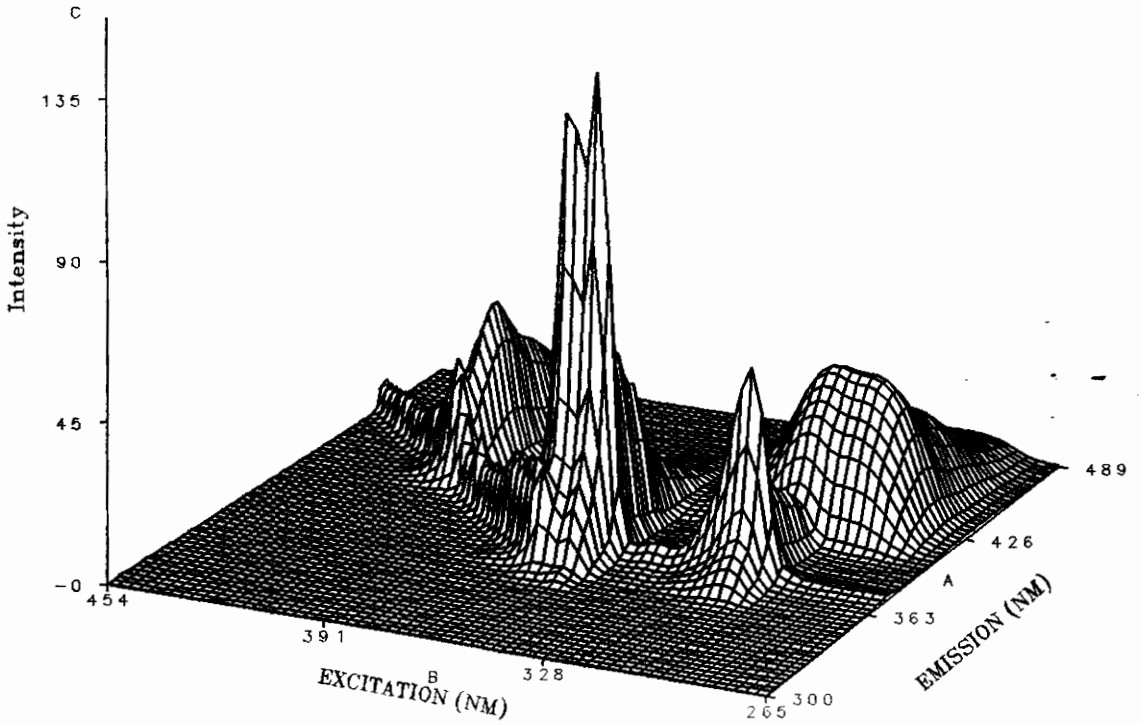
Mixture 3. (two components)



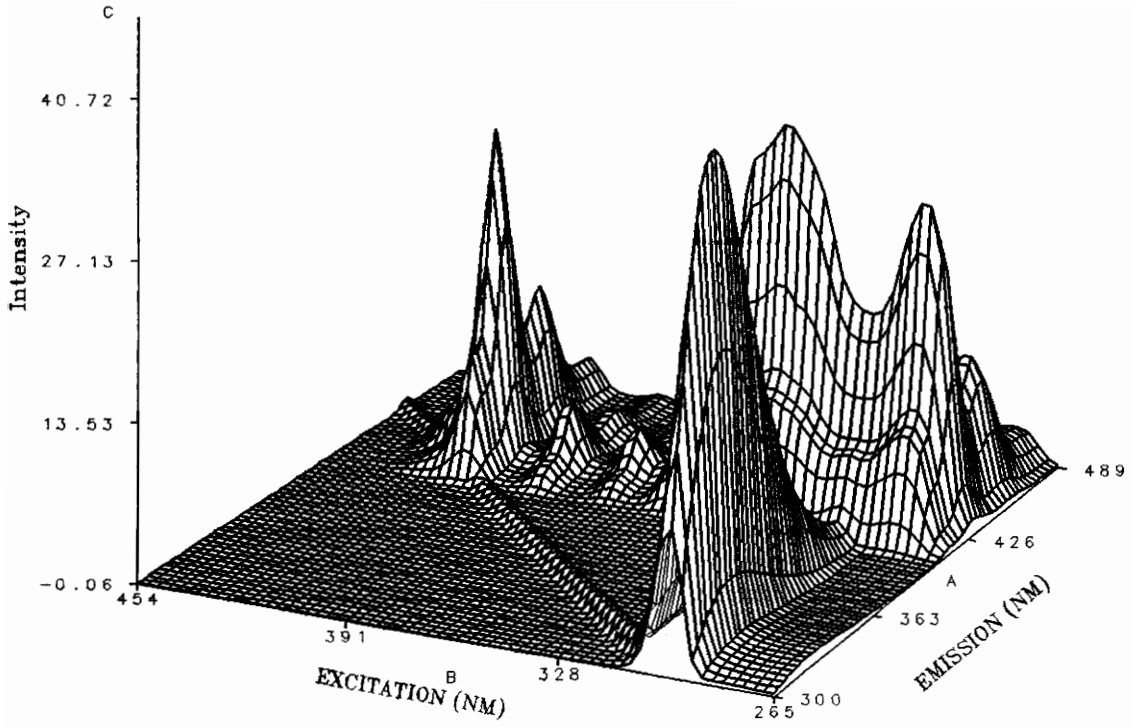
Mixture 4. (two components)



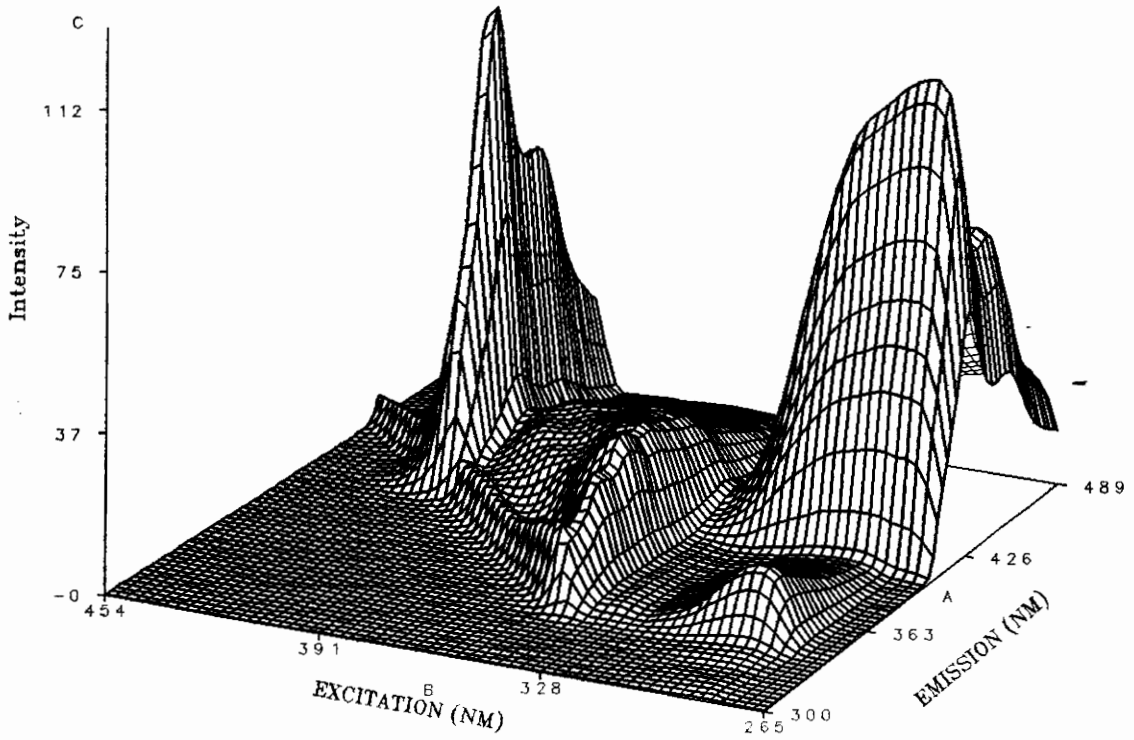
Mixture 5. (two components)



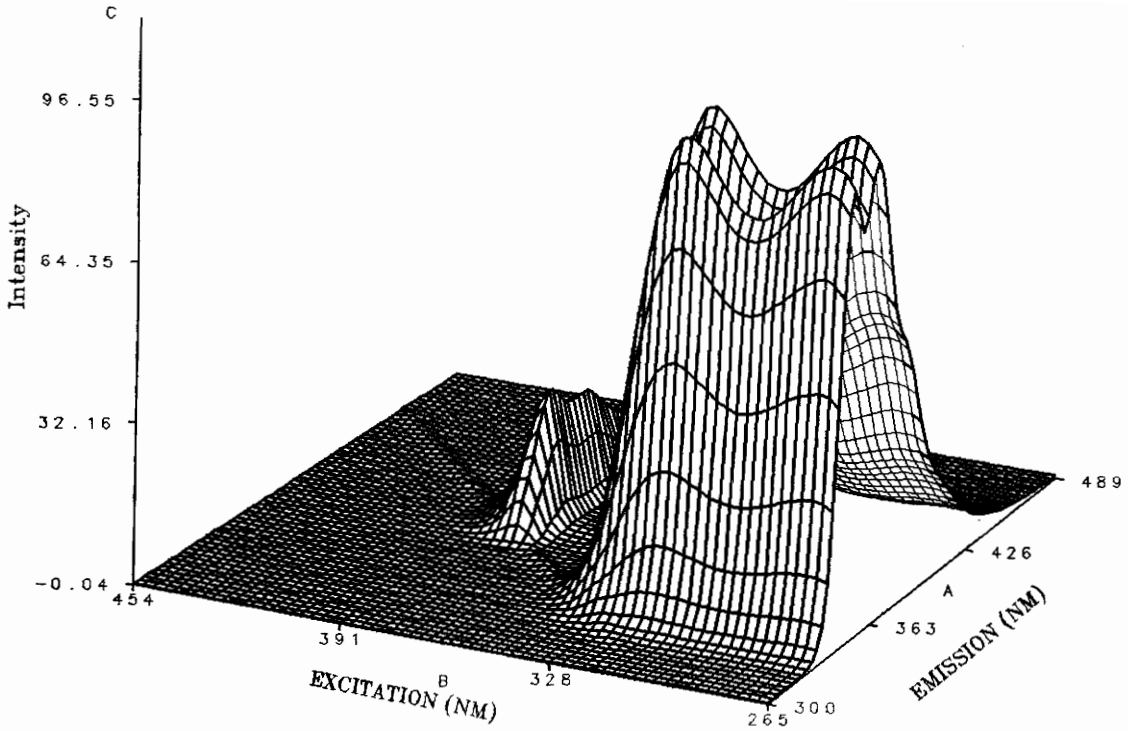
Mixture 6. (two components)



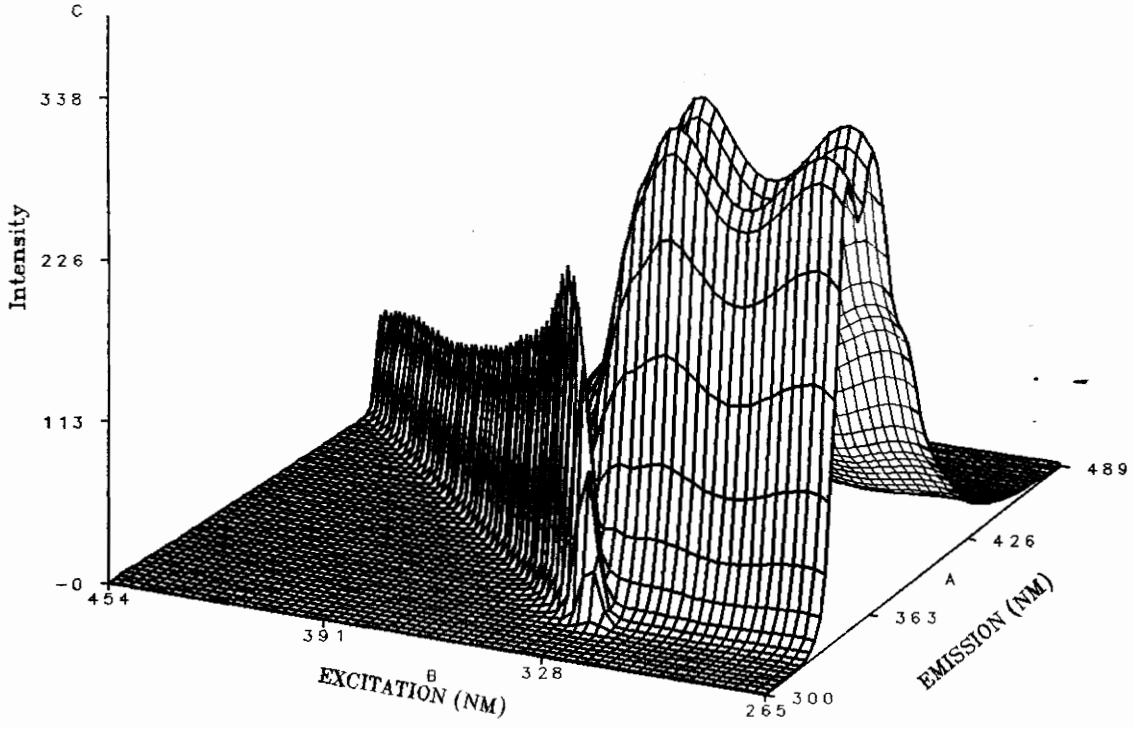
Mixture 7. (two components)



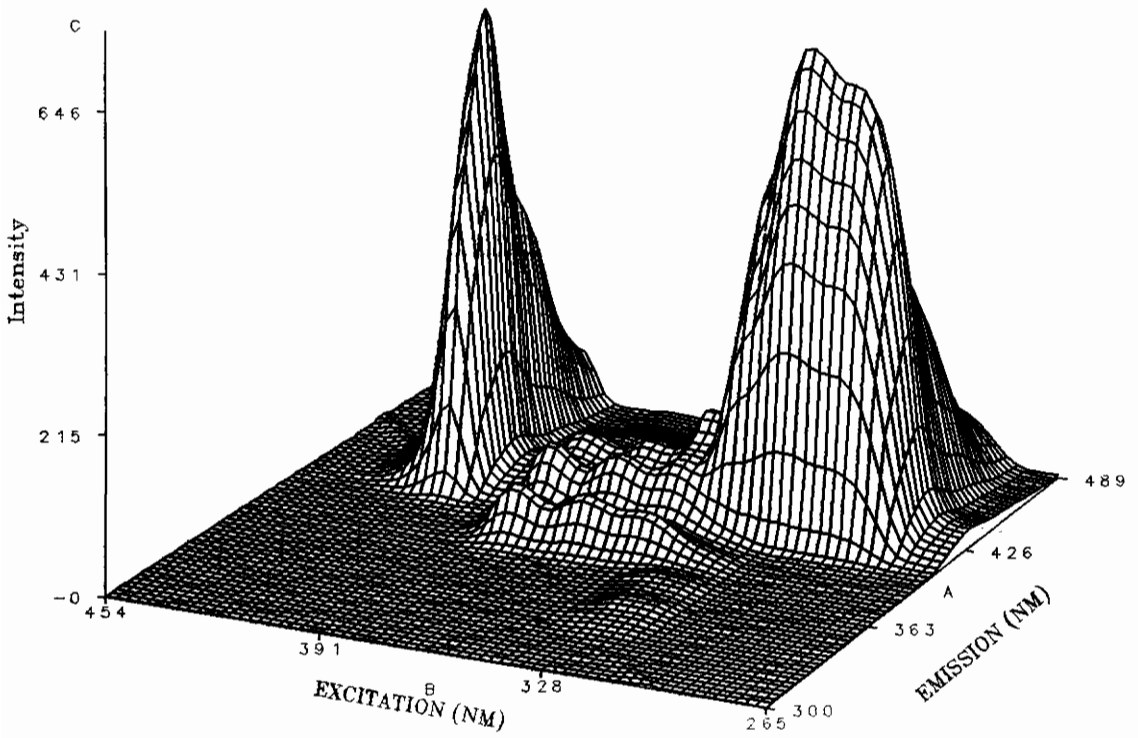
Mixture 8. (two components)



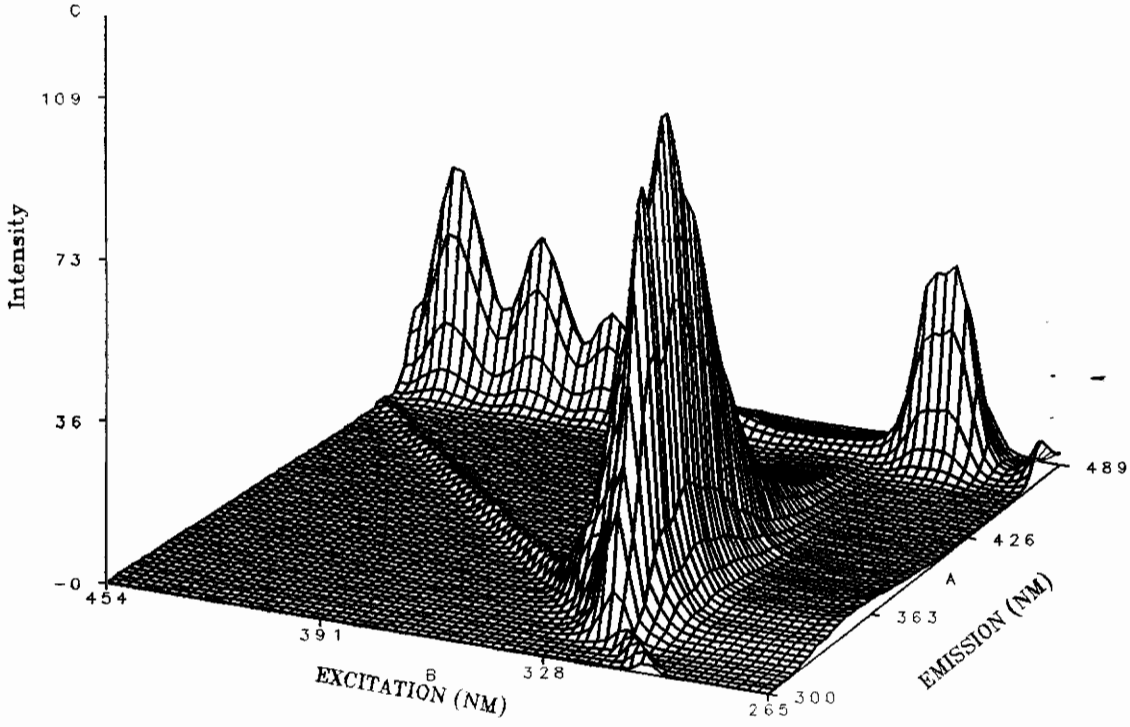
Mixture 9. (two components)



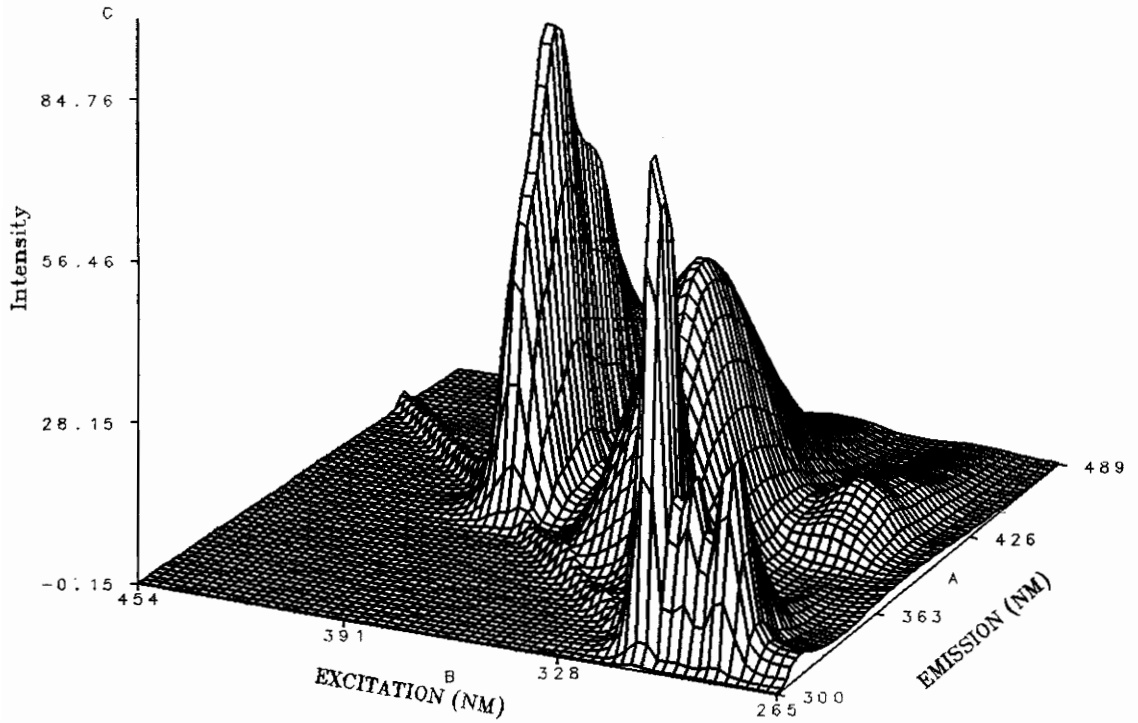
Mixture 10. (two components)



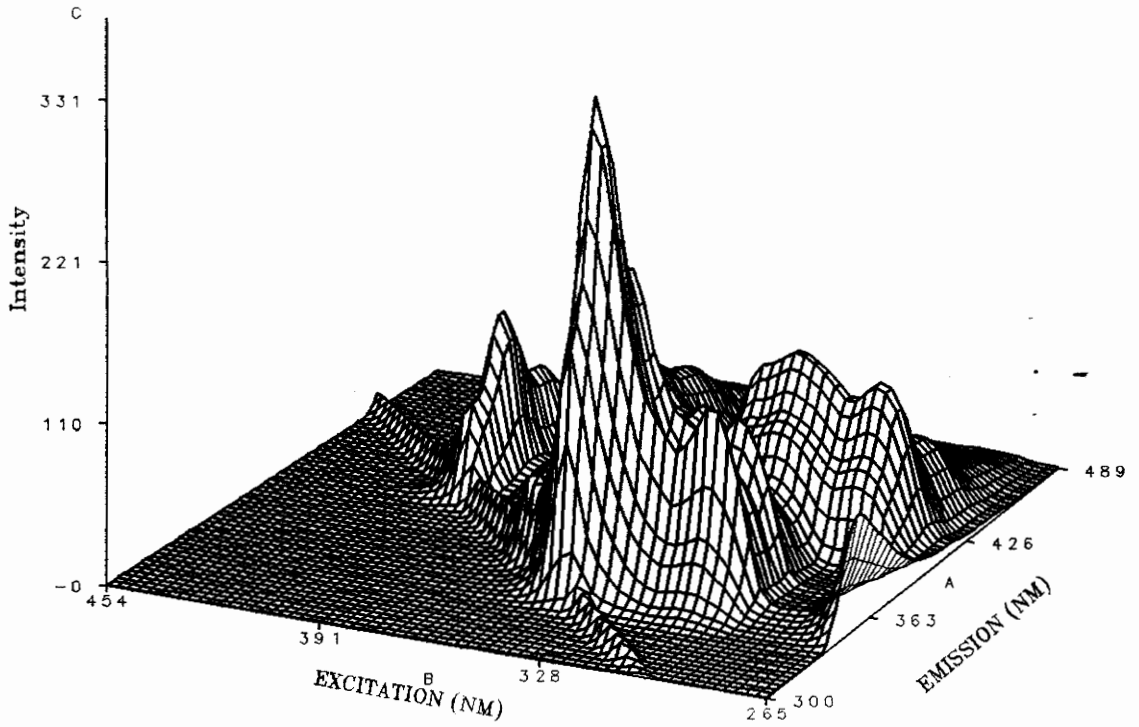
Mixture 11. (three components)



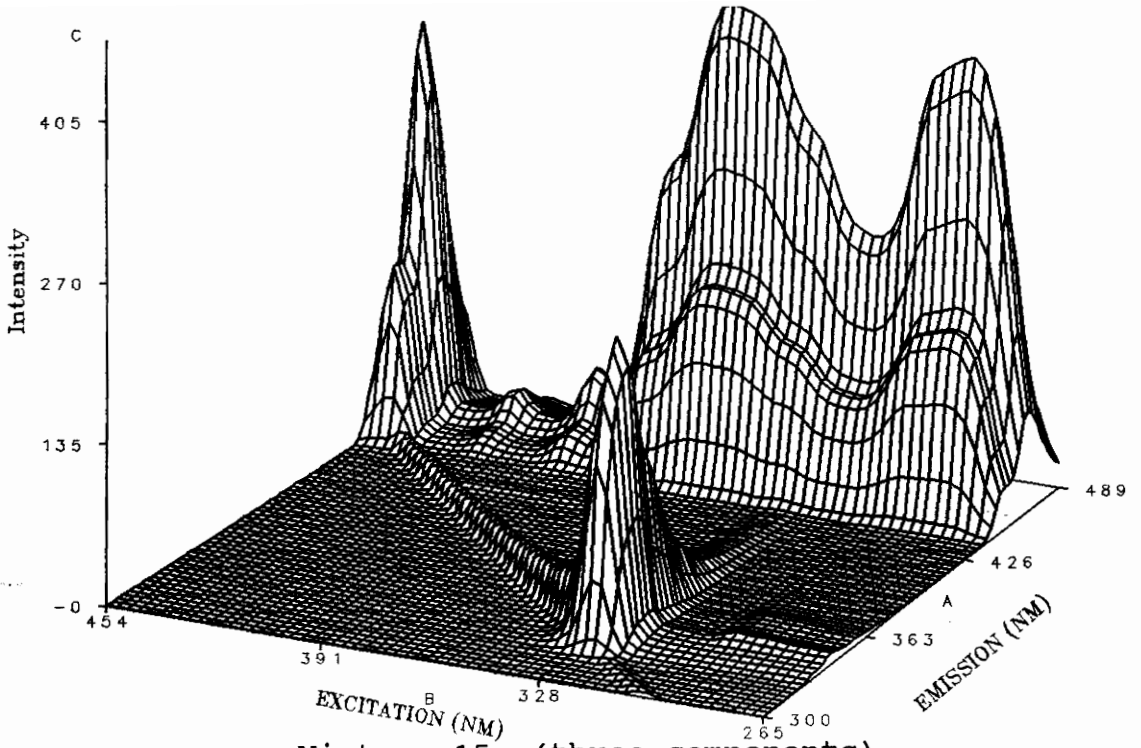
Mixture 12. (three components)



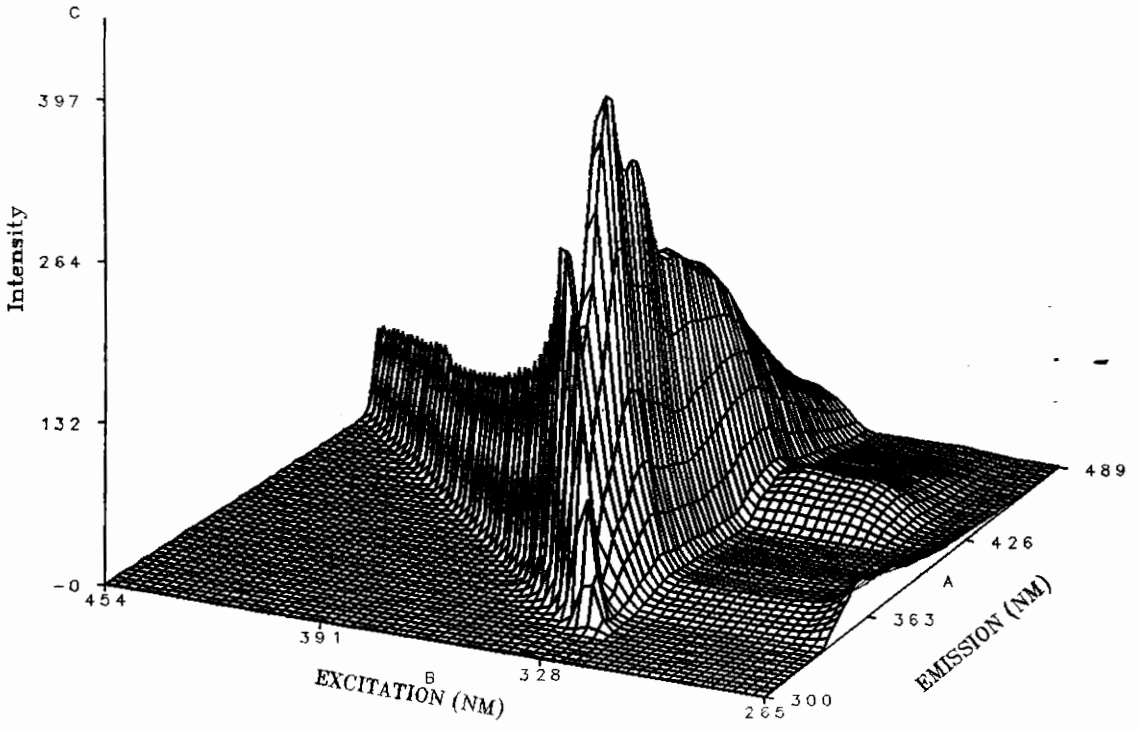
Mixture 13. (three components)



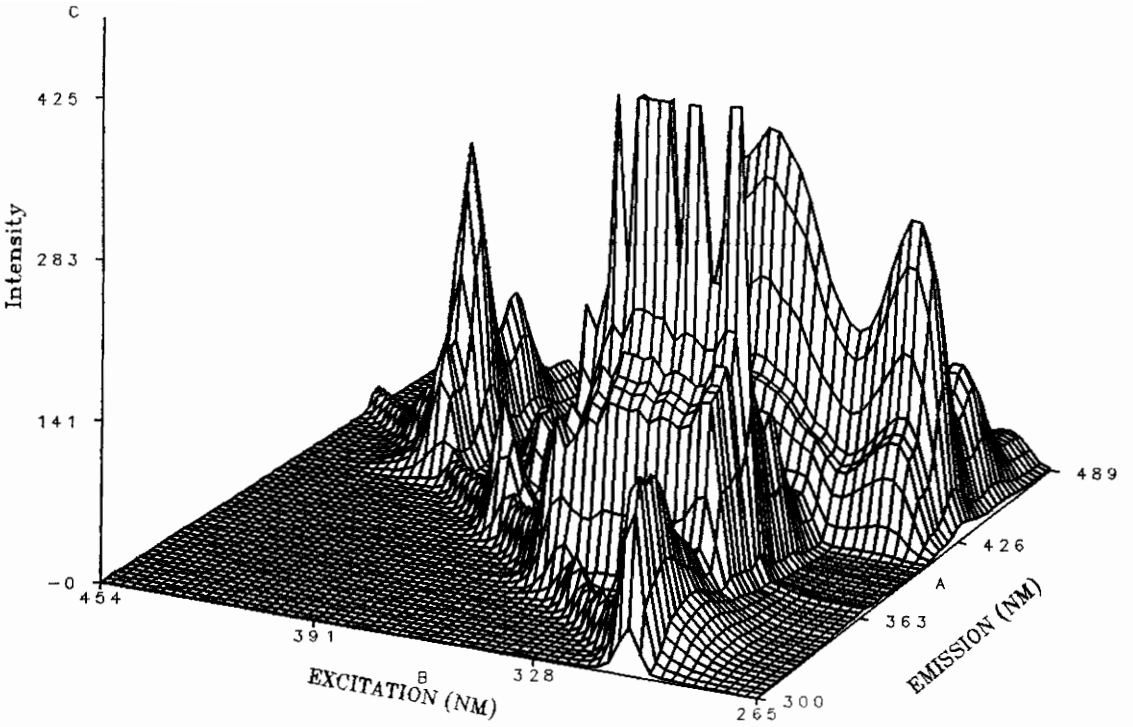
Mixture 14. (three components)



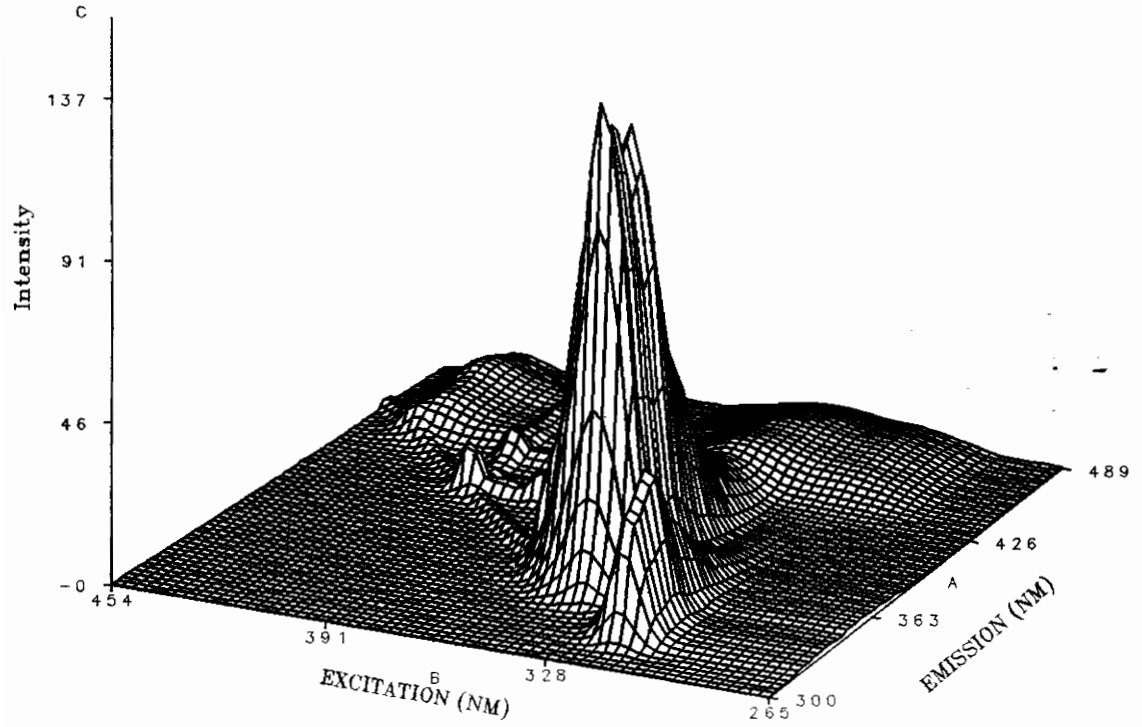
Mixture 15. (three components)



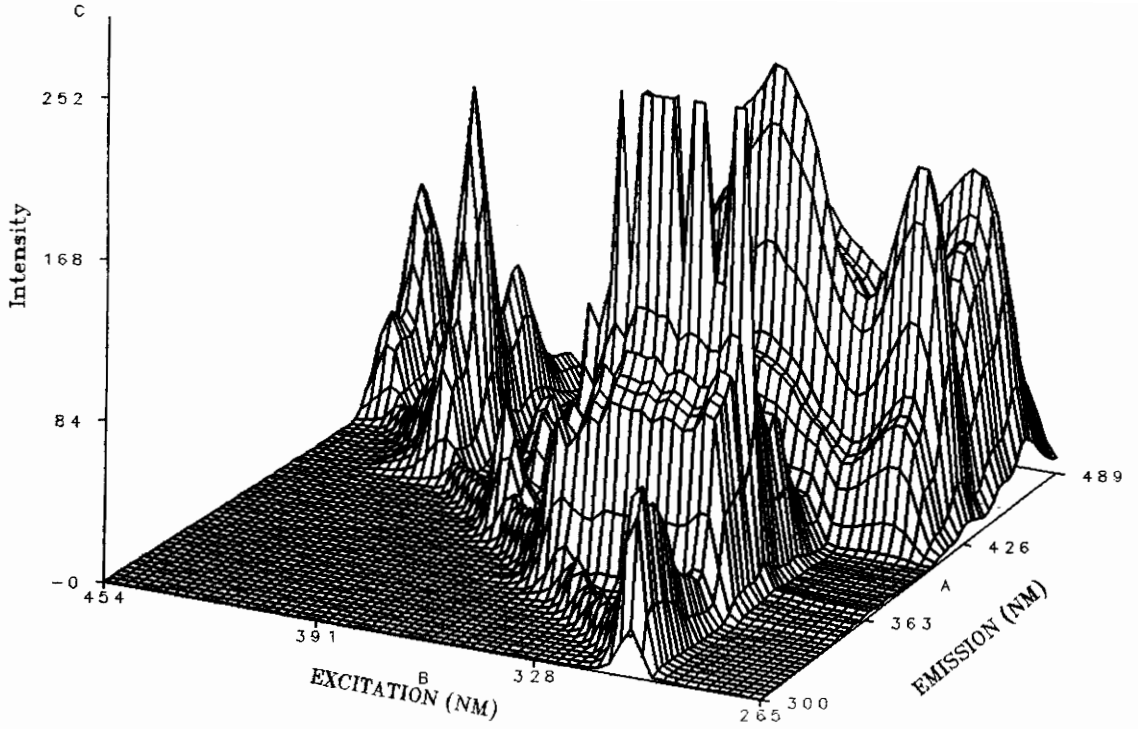
Mixture 16. (three components)



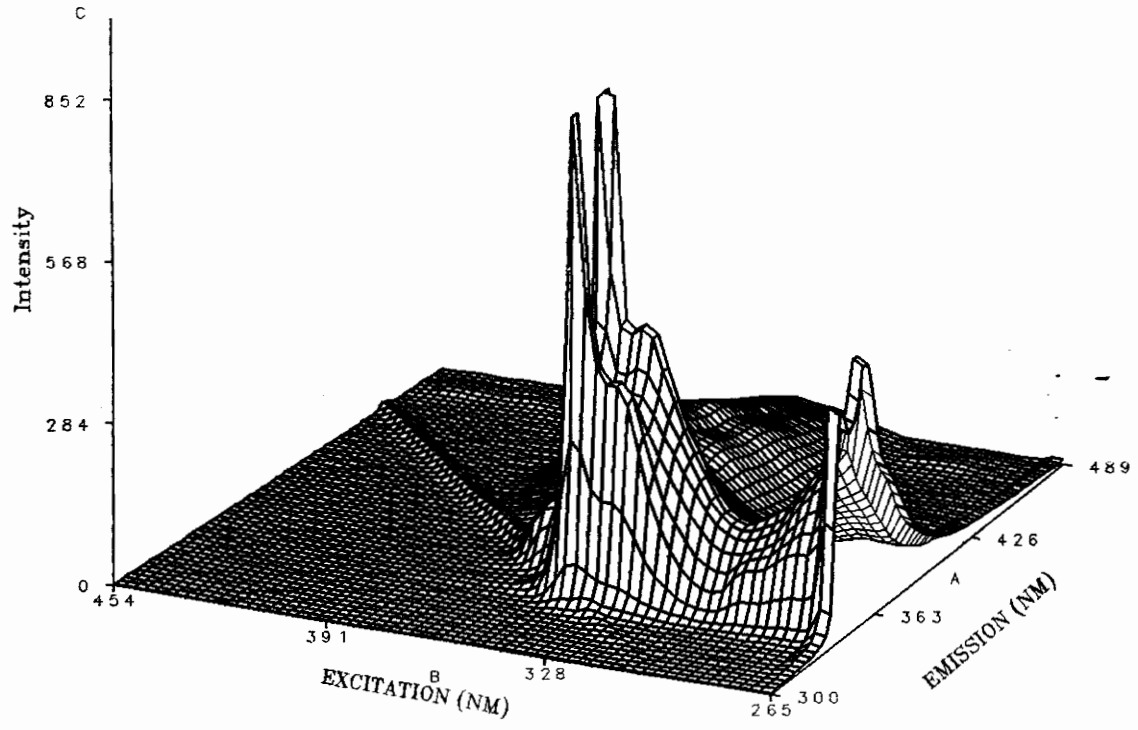
Mixture 17. (three components)



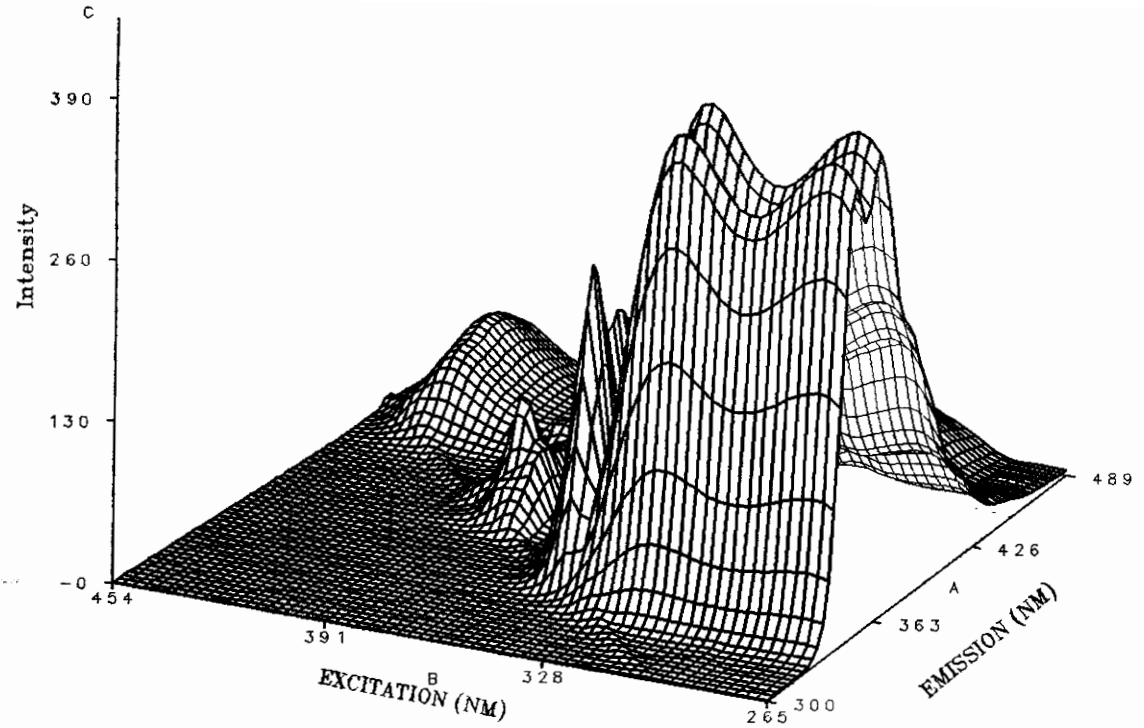
Mixture 18. (three components)



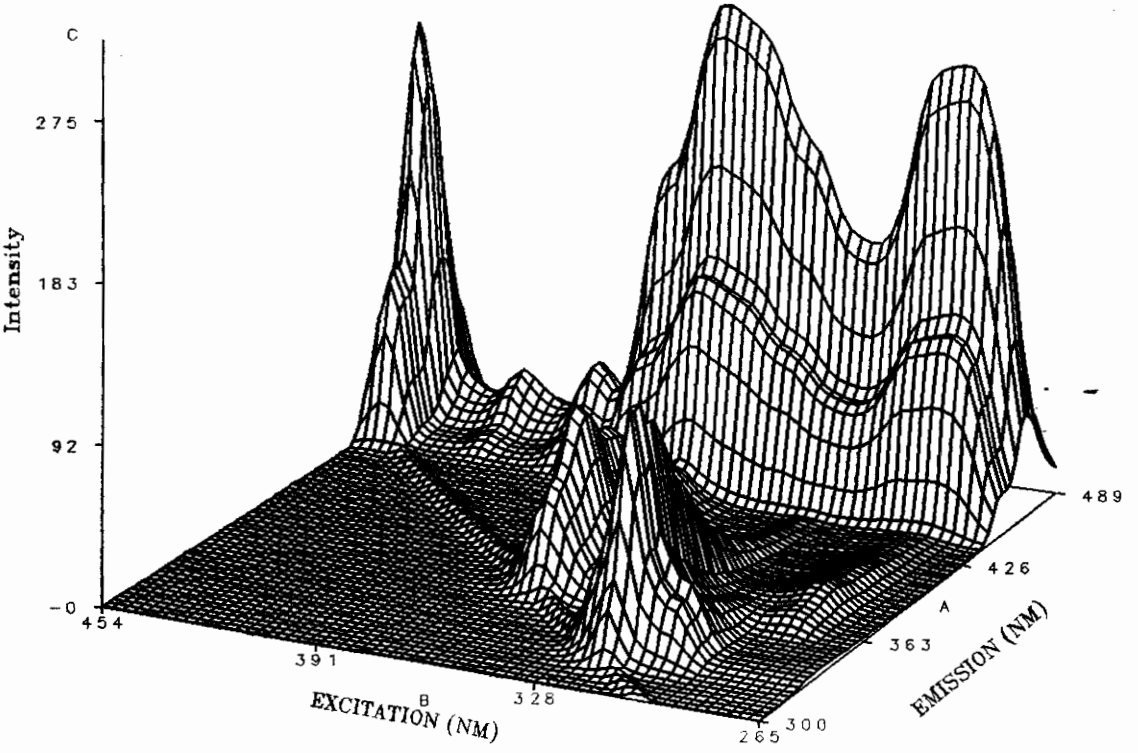
Mixture 19. (three components)



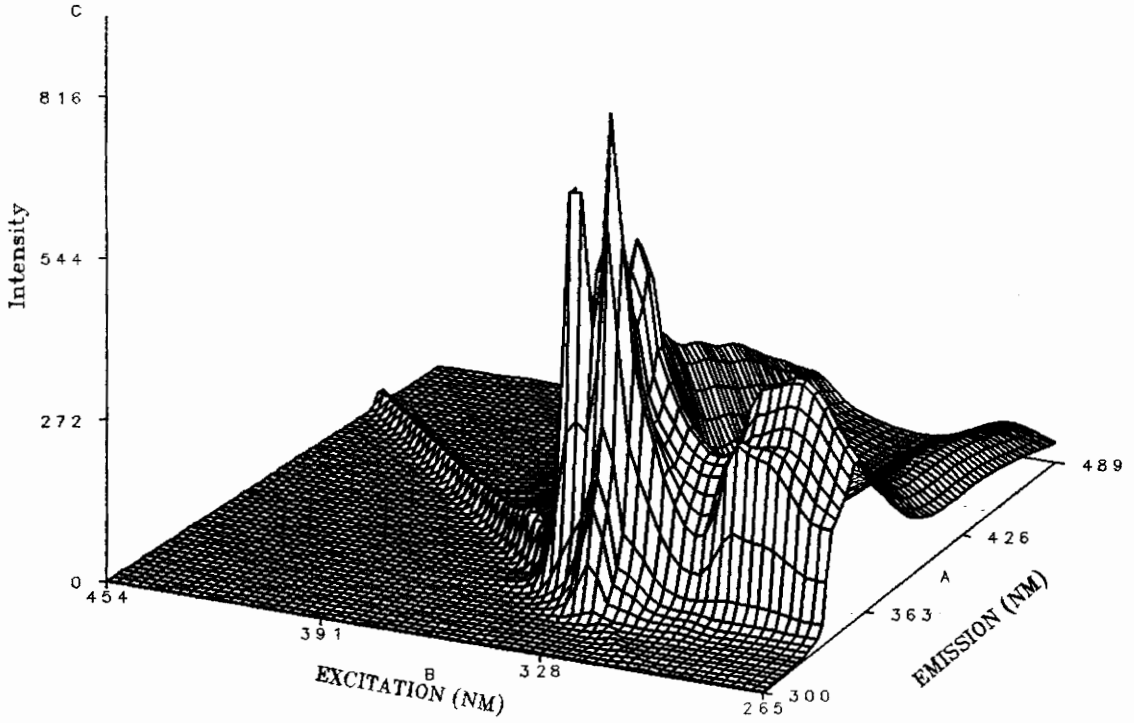
Mixture 20. (three components)



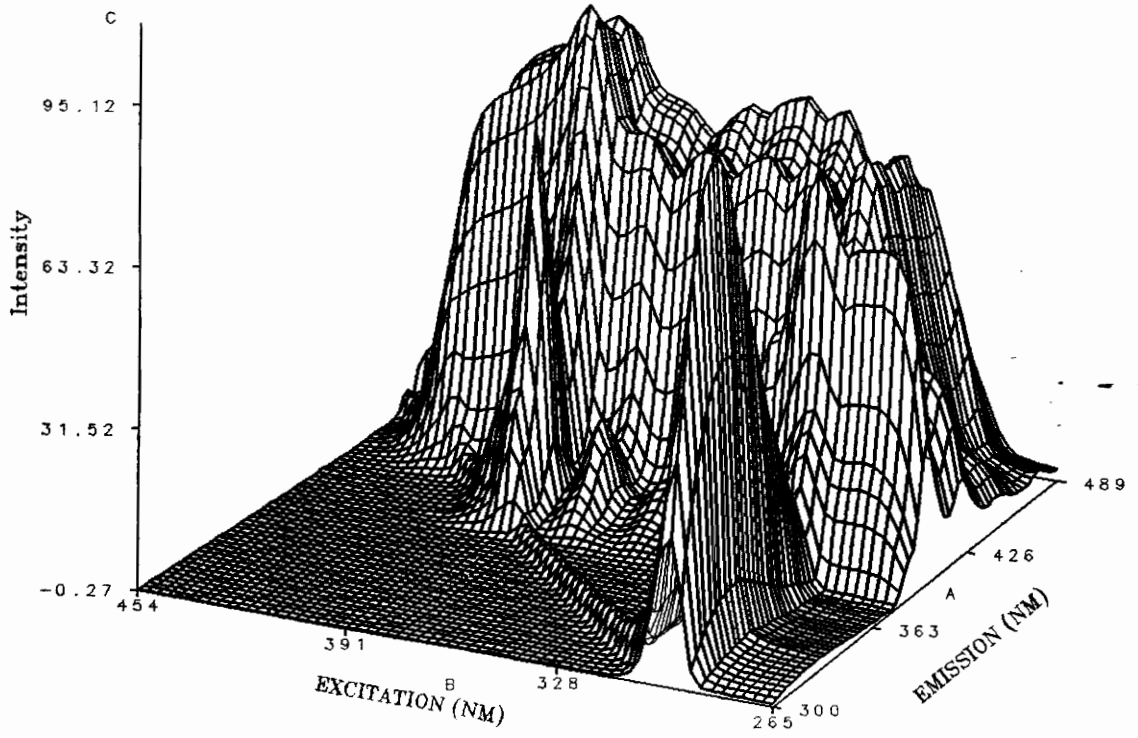
Mixture 21. (five components)



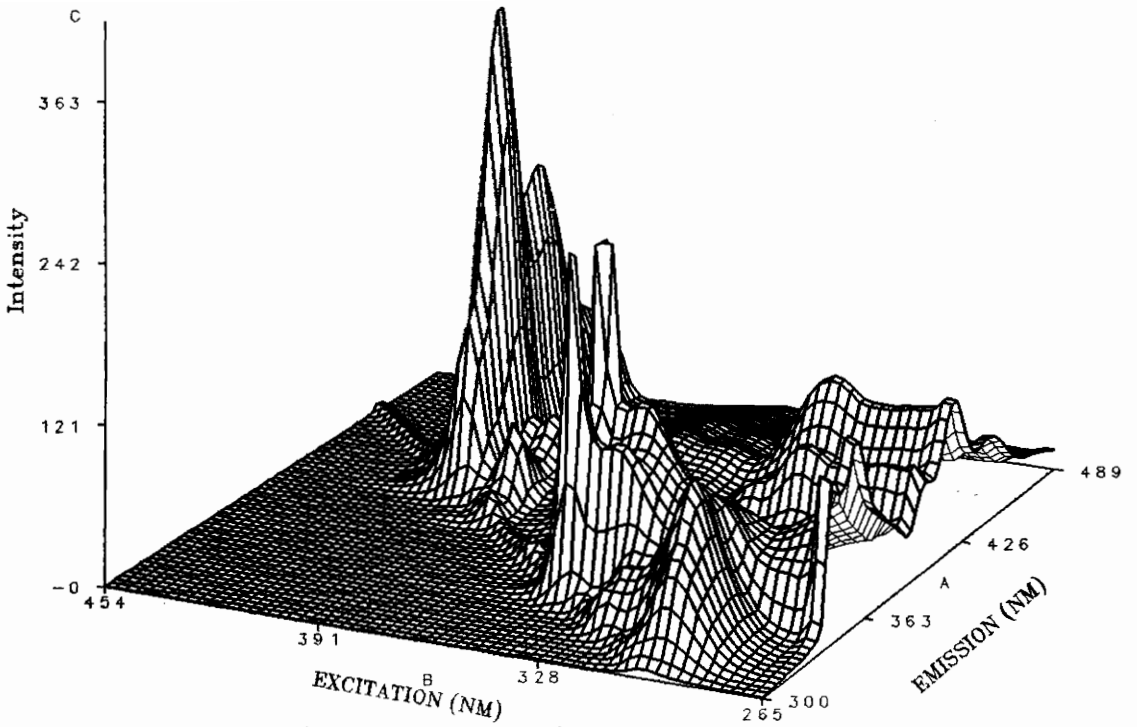
Mixture 22. (five components)



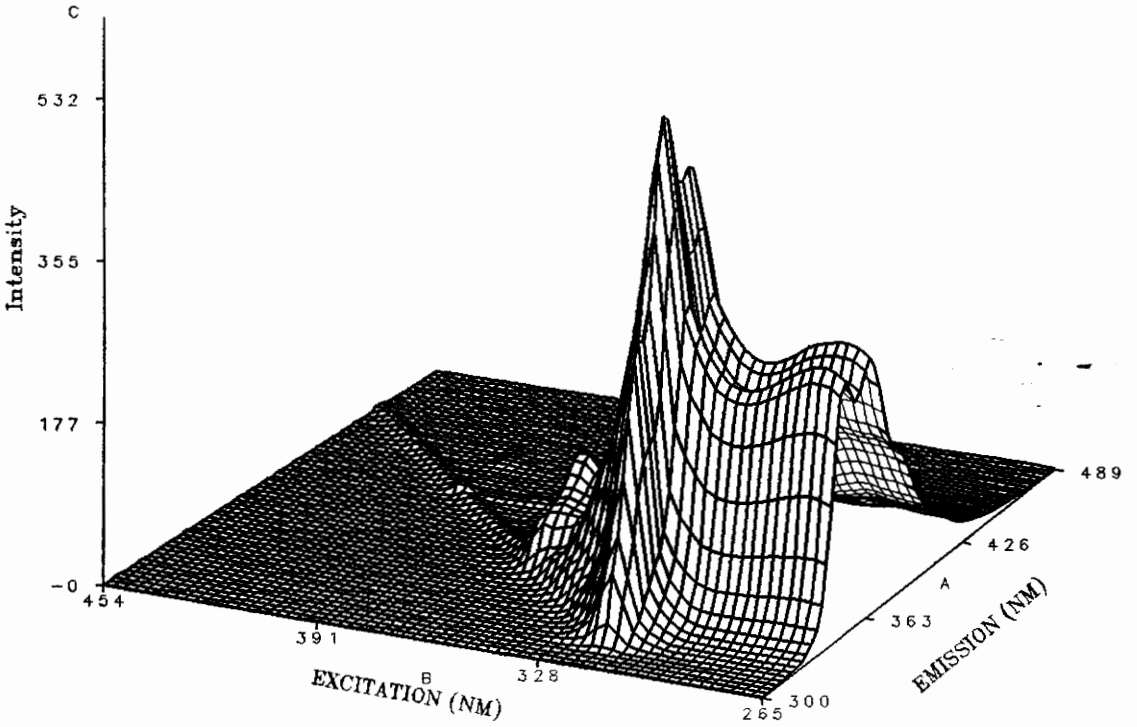
Mixture 23. (five components)



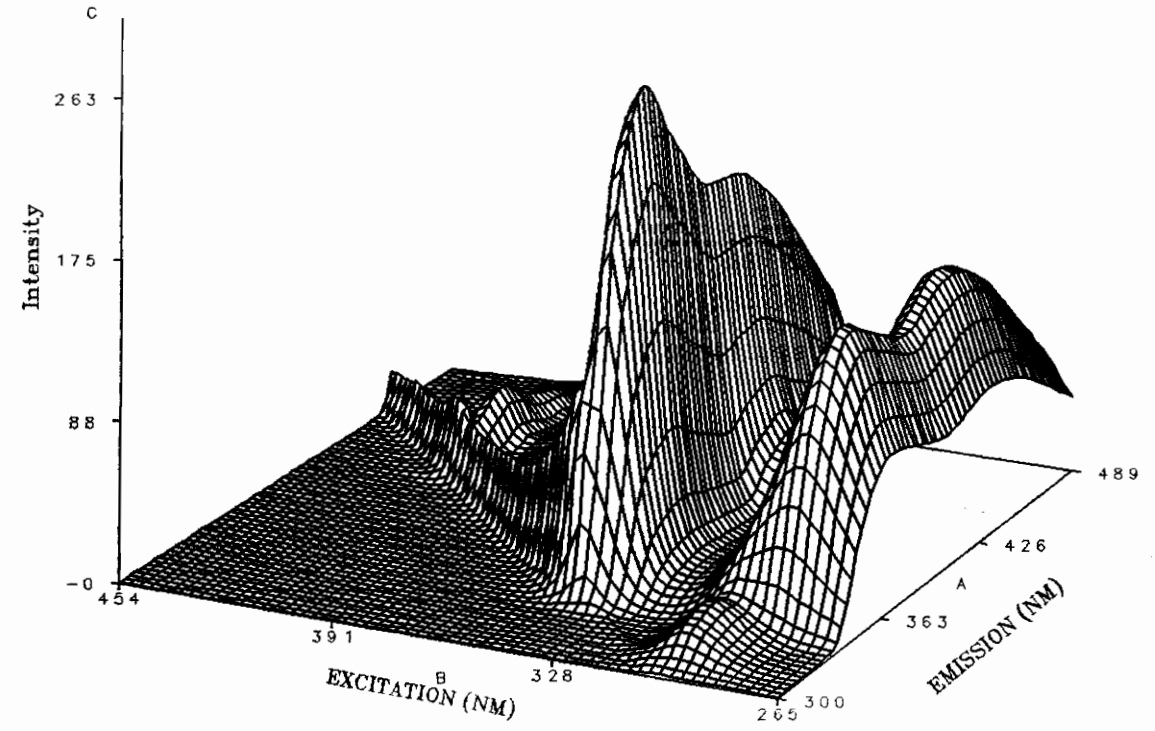
Mixture 24. (five components)



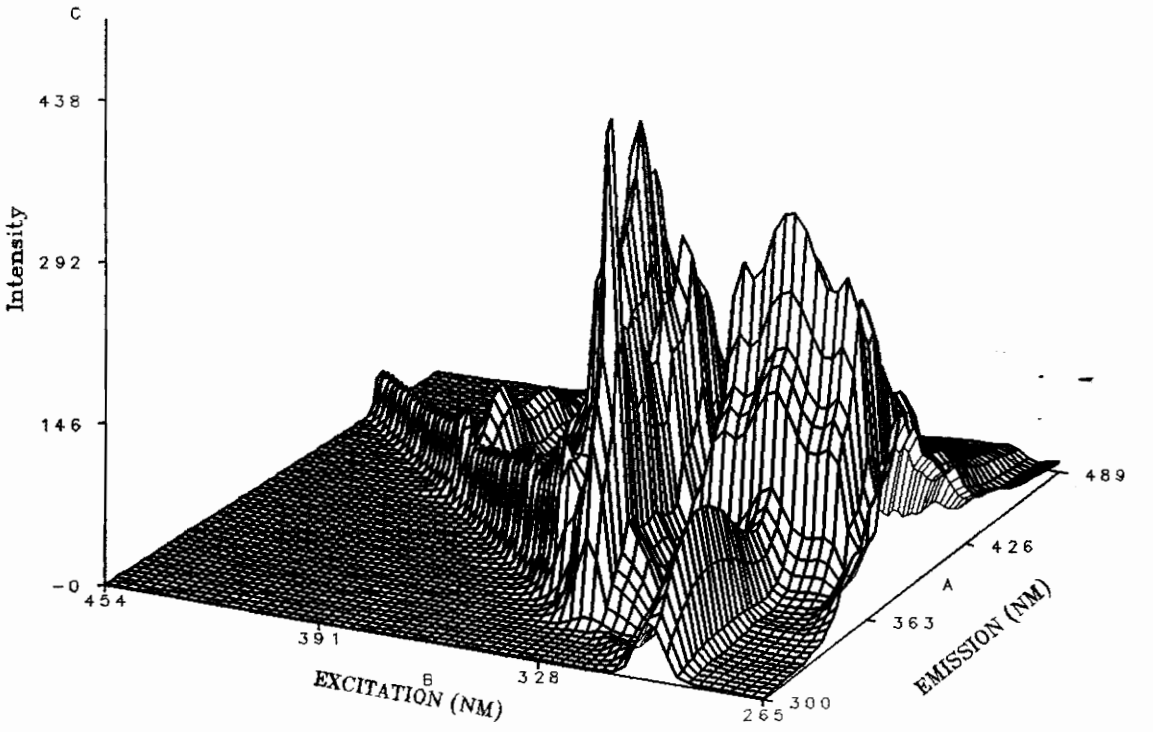
Mixture 25. (five components)



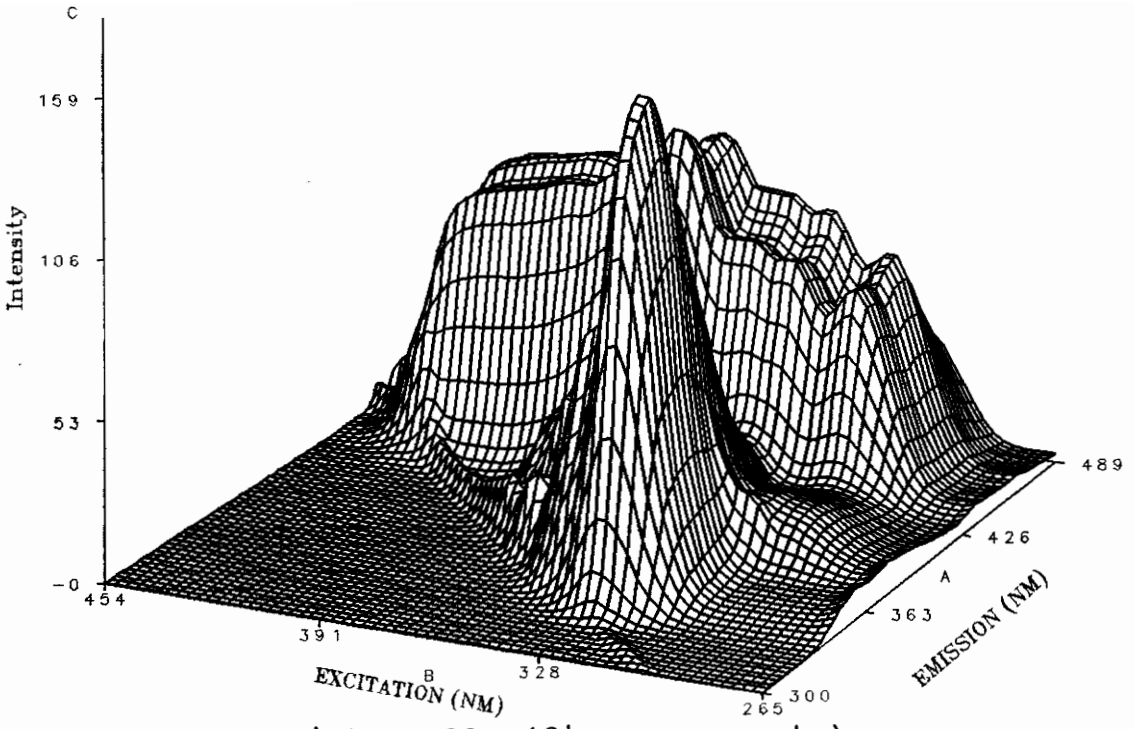
Mixture 26. (five components)



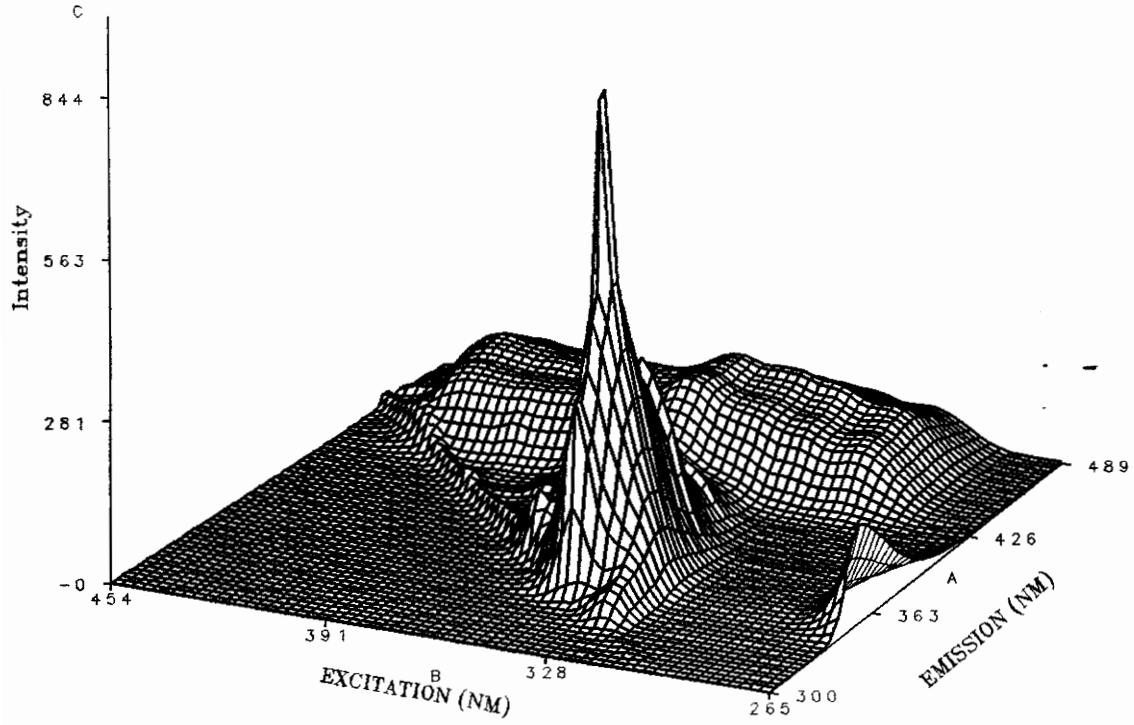
Mixture 27. (five components)



Mixture 28. (five components)



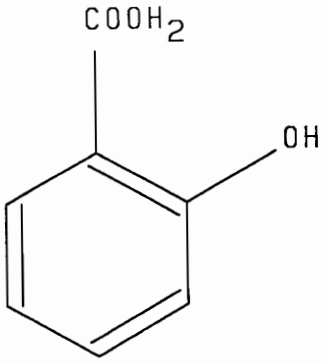
Mixture 29. (five components)



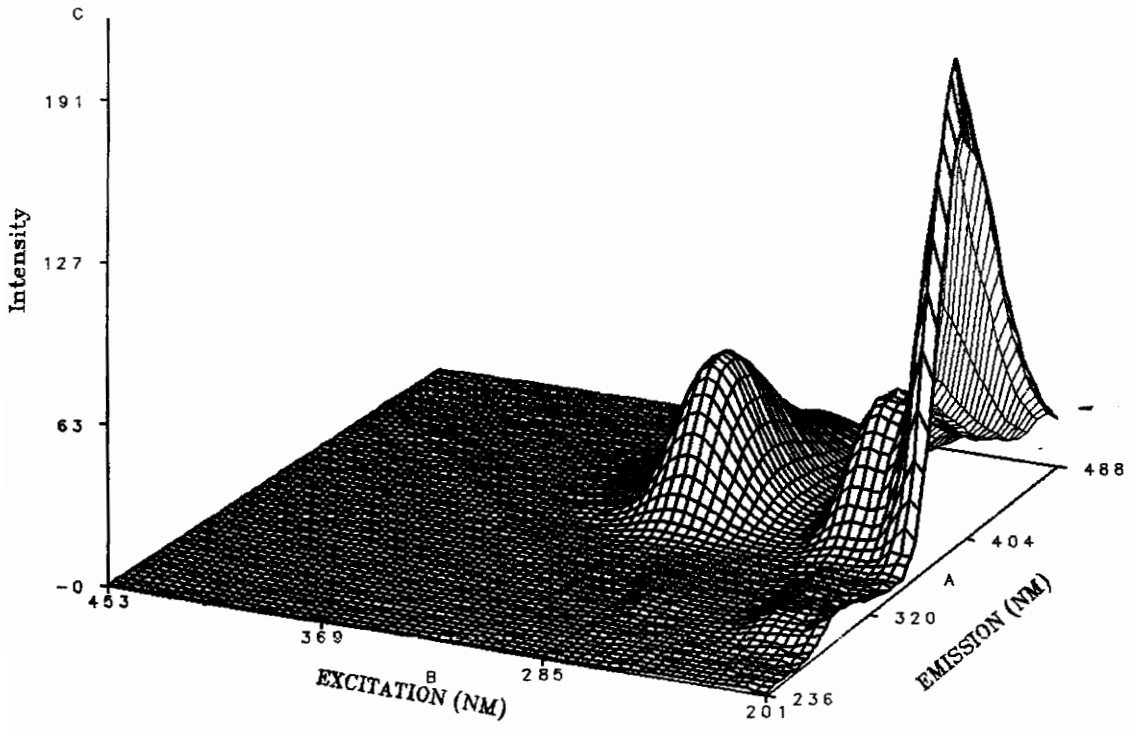
Mixture 30. (five components)

Appendix D.
Reference library.
Molecular Structures and Excitation Emission Matrices.

a)

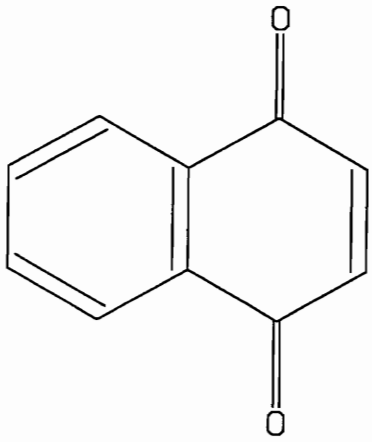


b)

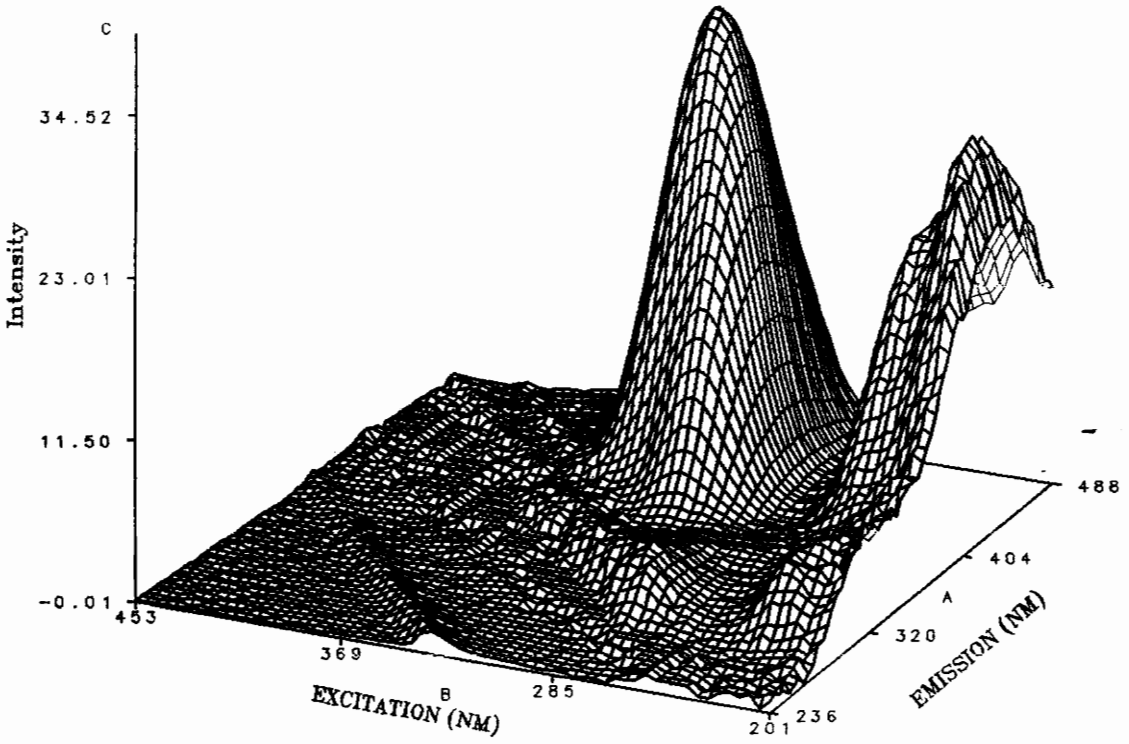


Salicylic Acid. a) Structure
b) EEM

a)

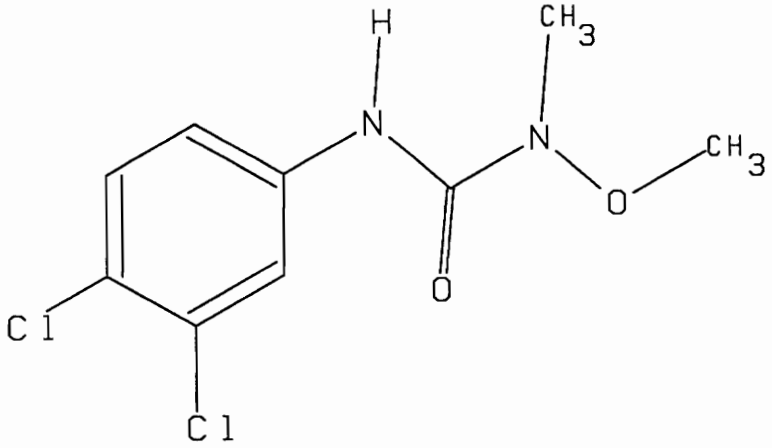


b)

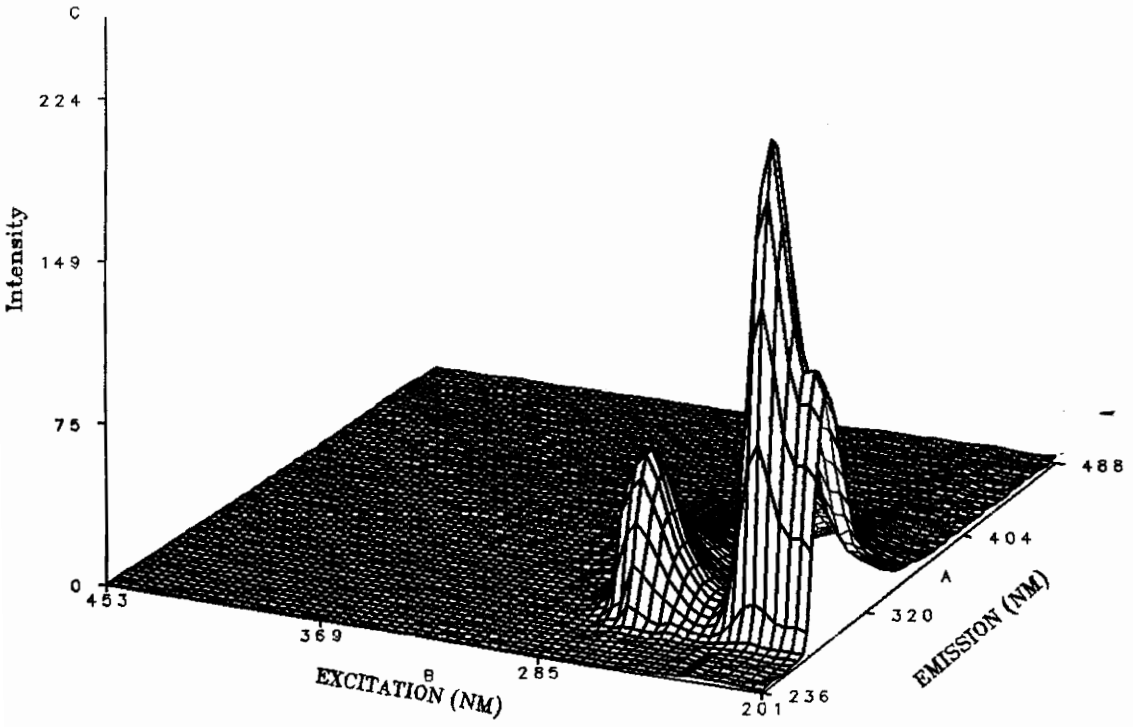


1,4-Naphthoquinone. a) Structure
b) EEM

a)

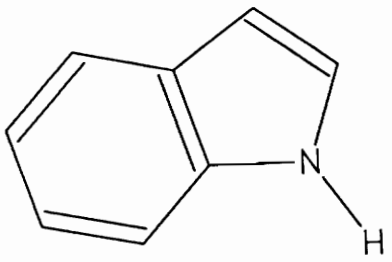


b)

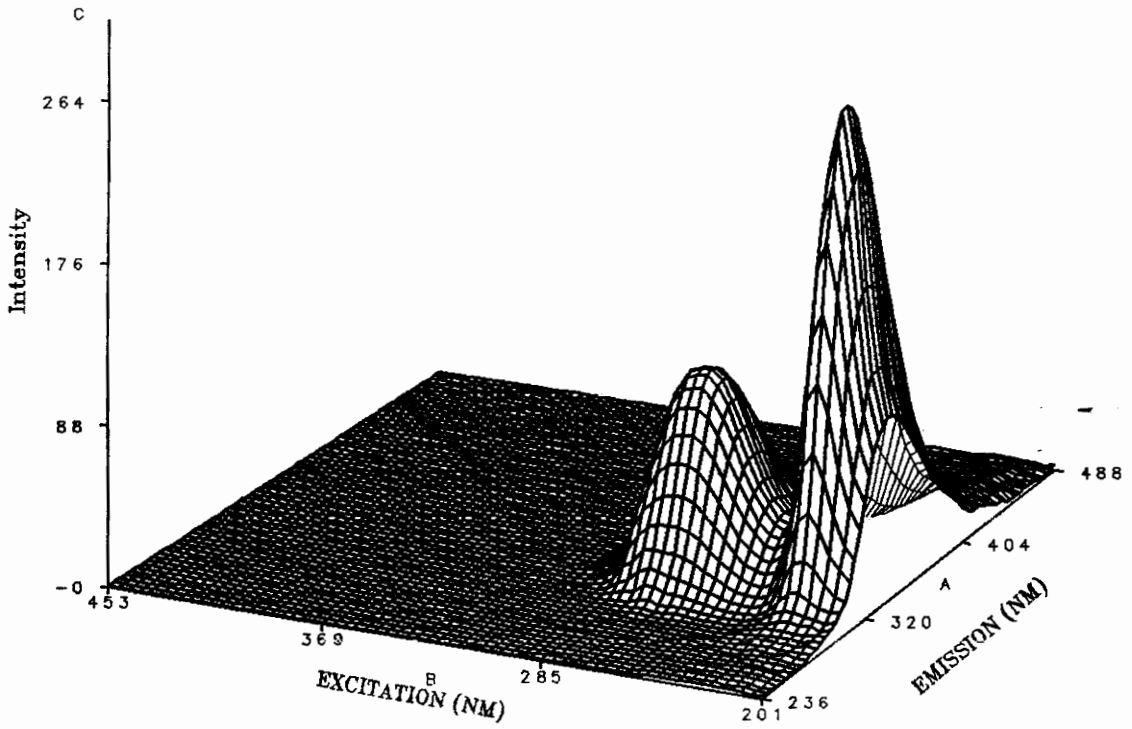


Linuron. a) Structure
b) EEM

a)

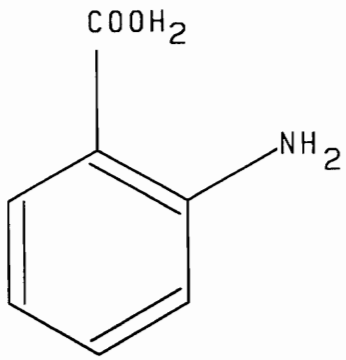


b)

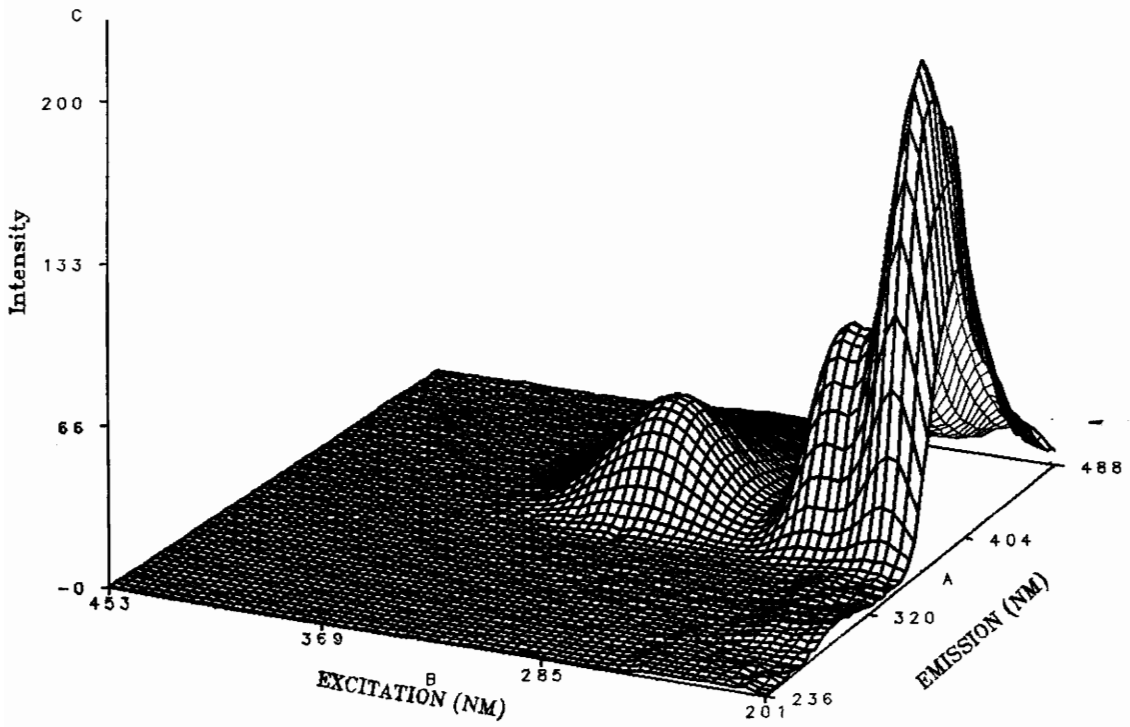


Indole. a) Structure
b) EEM

a)

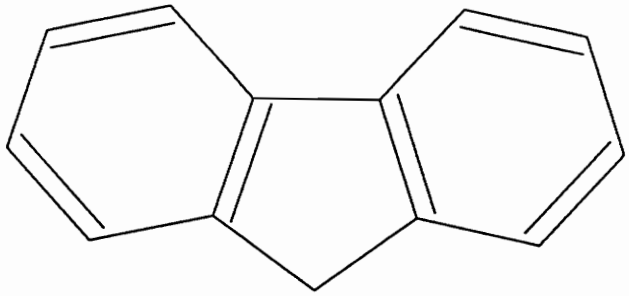


b)

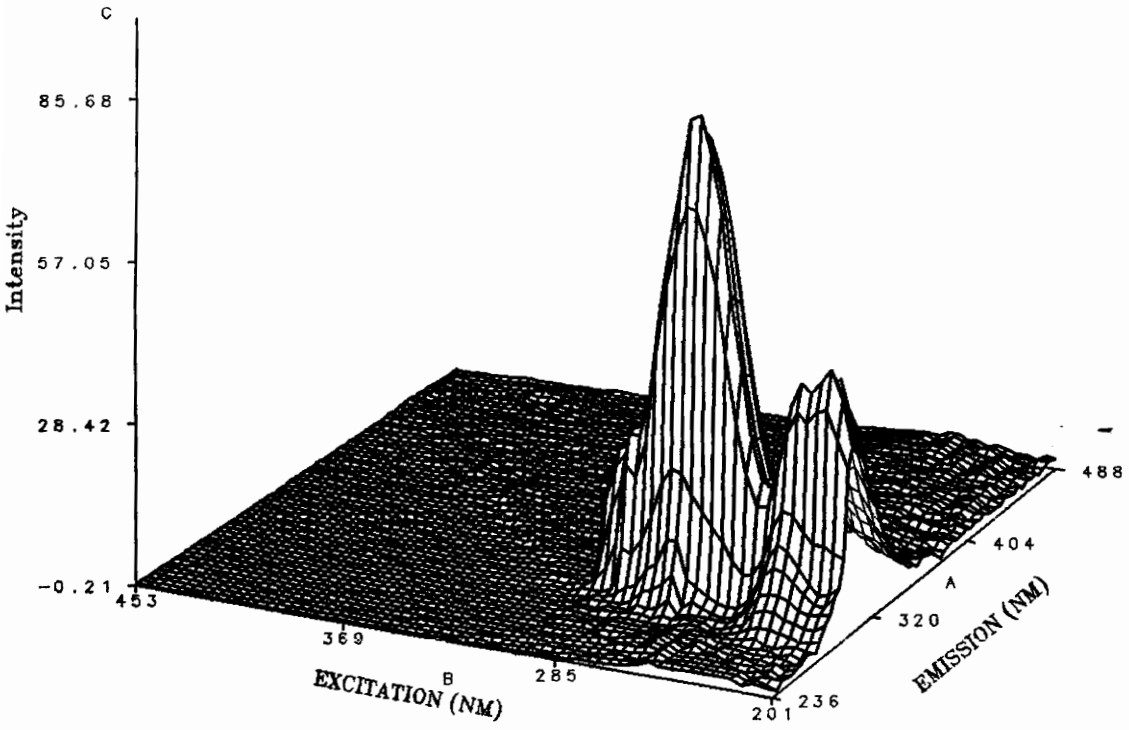


Anthranilic Acid. a) Structure
b) EEM

a)

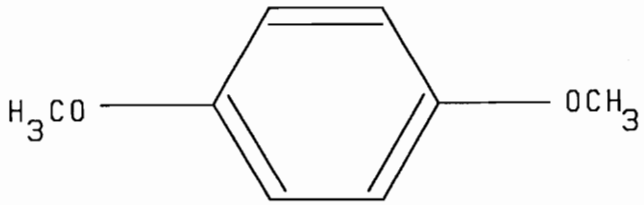


b)

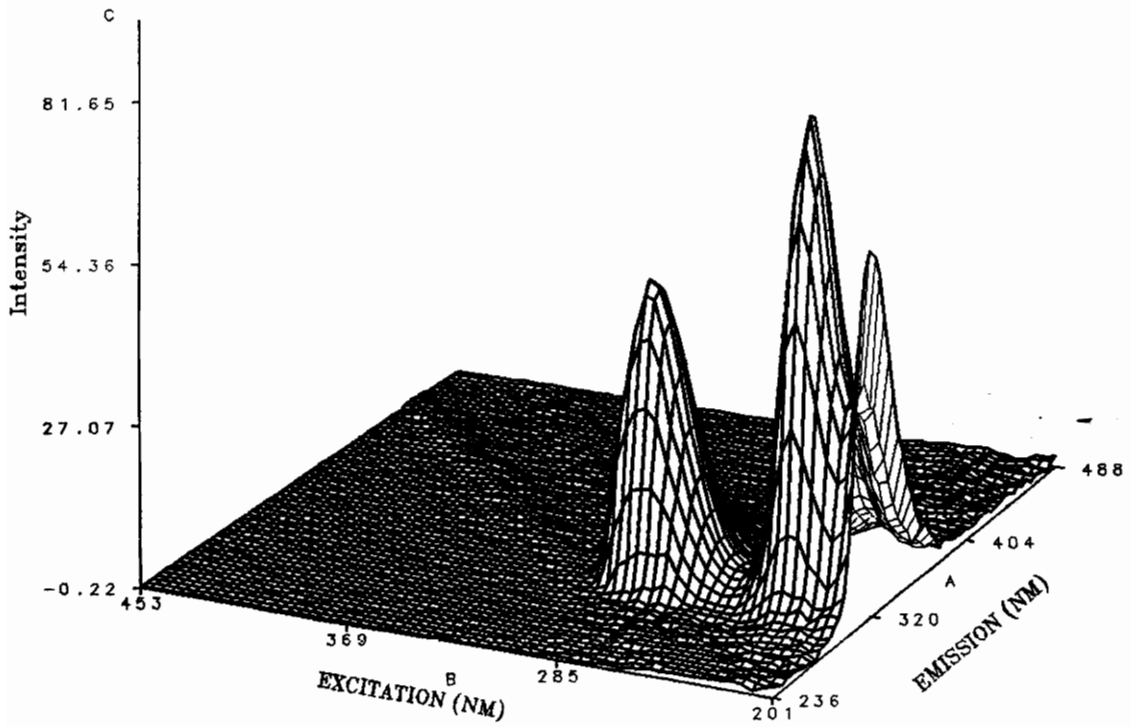


Fluorene. a) Structure
b) EEM

a)

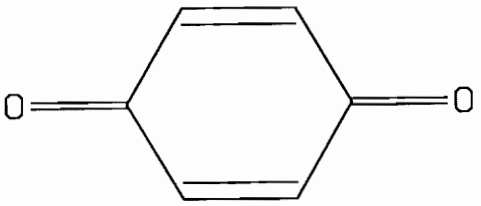


b)

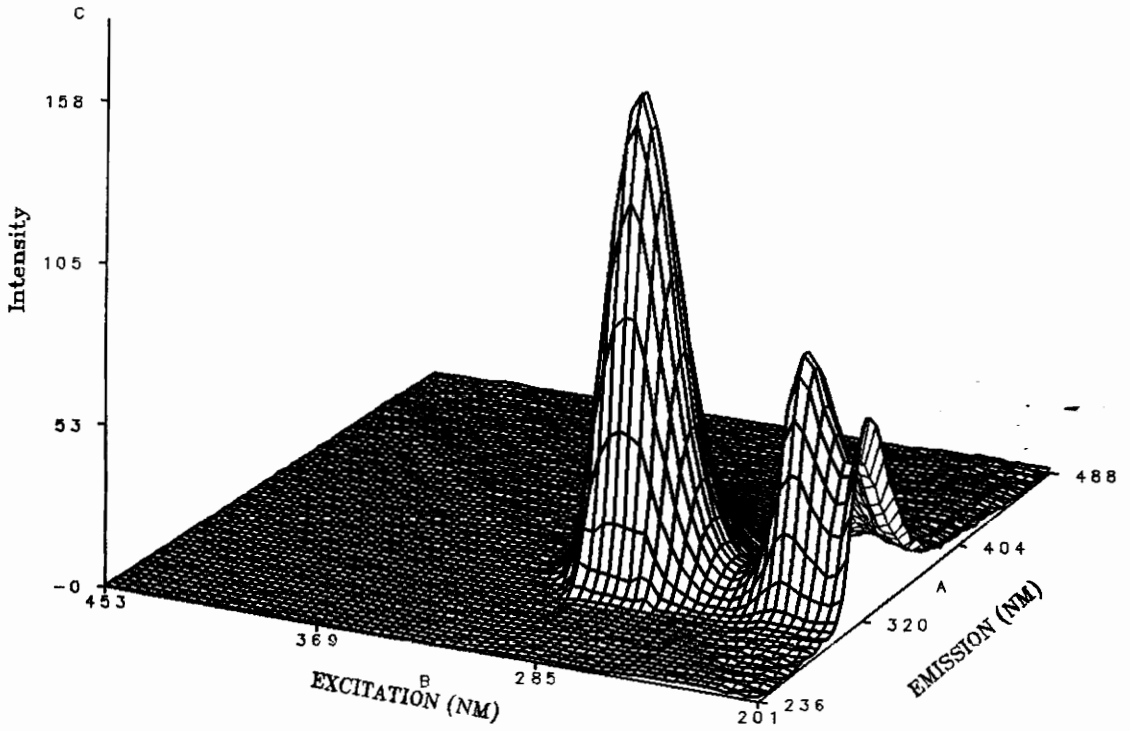


1,4-Dimethoxybenzene. a) Structure
b) EEM

a)

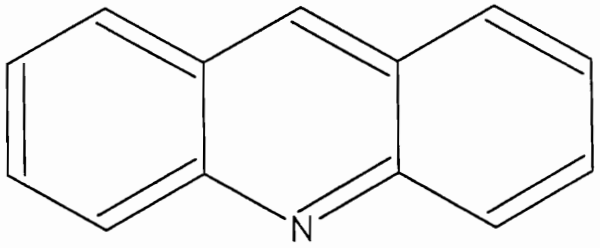


b)

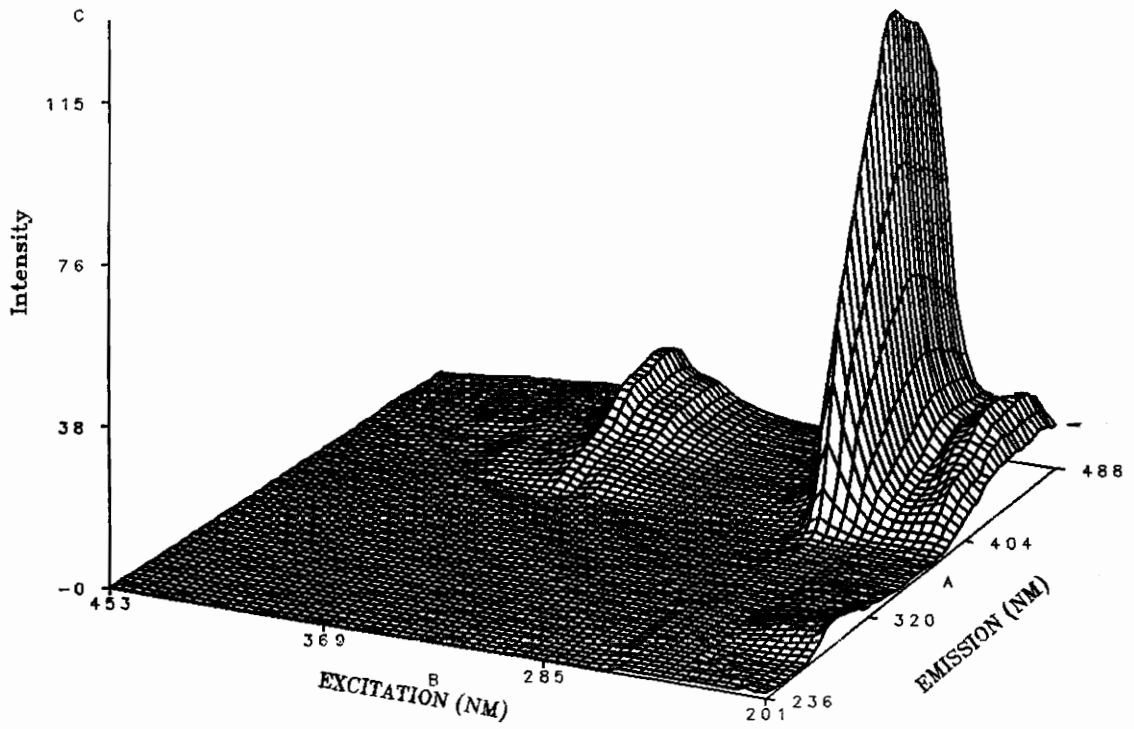


Benzoquinone. a) Structure
b) EEM

a)

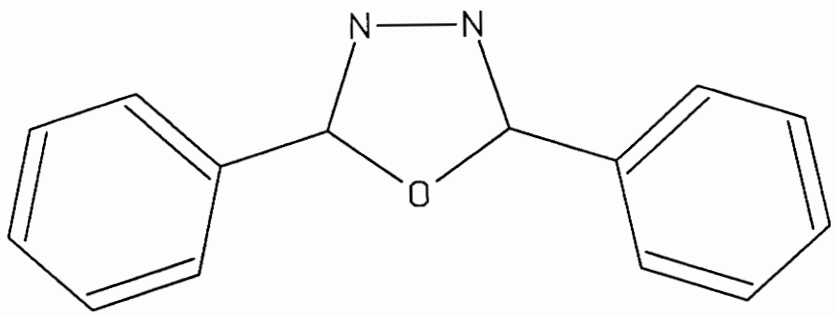


b)

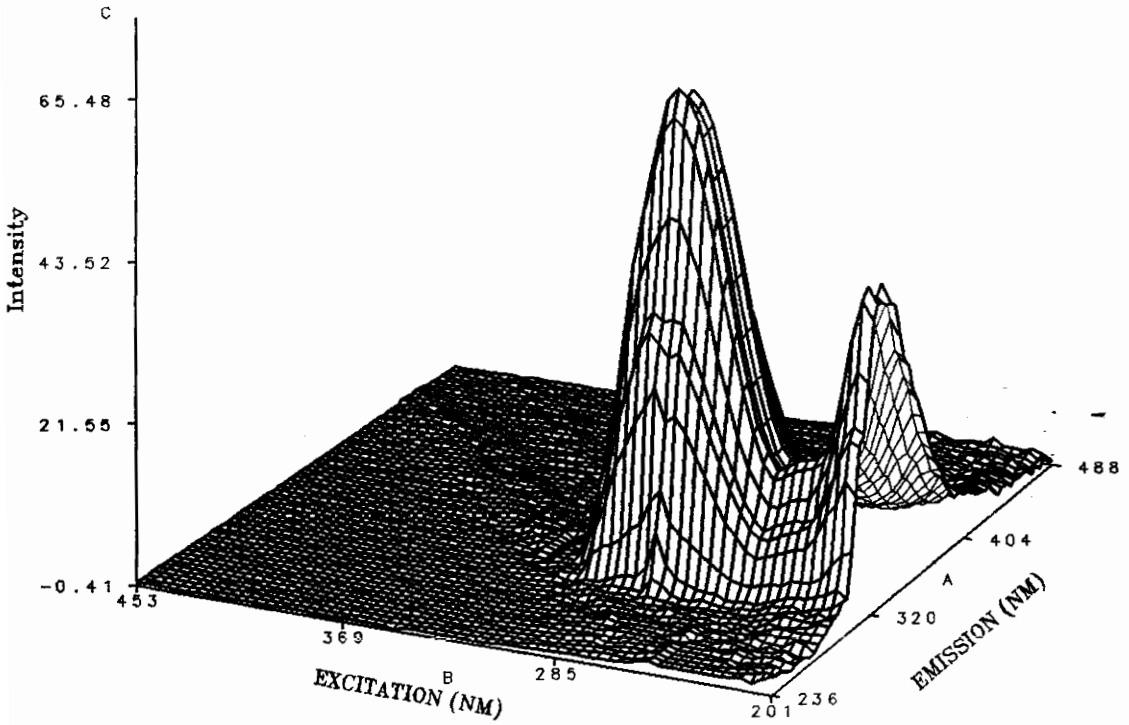


Acridine. a) Structure
b) EEM

a)

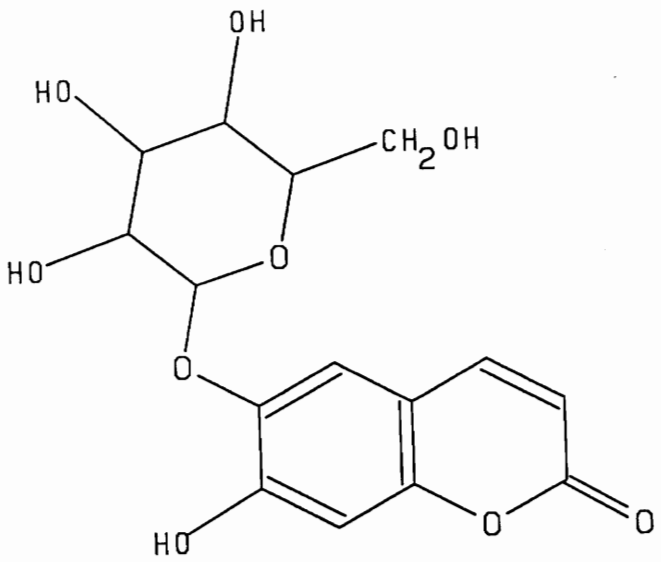


b)

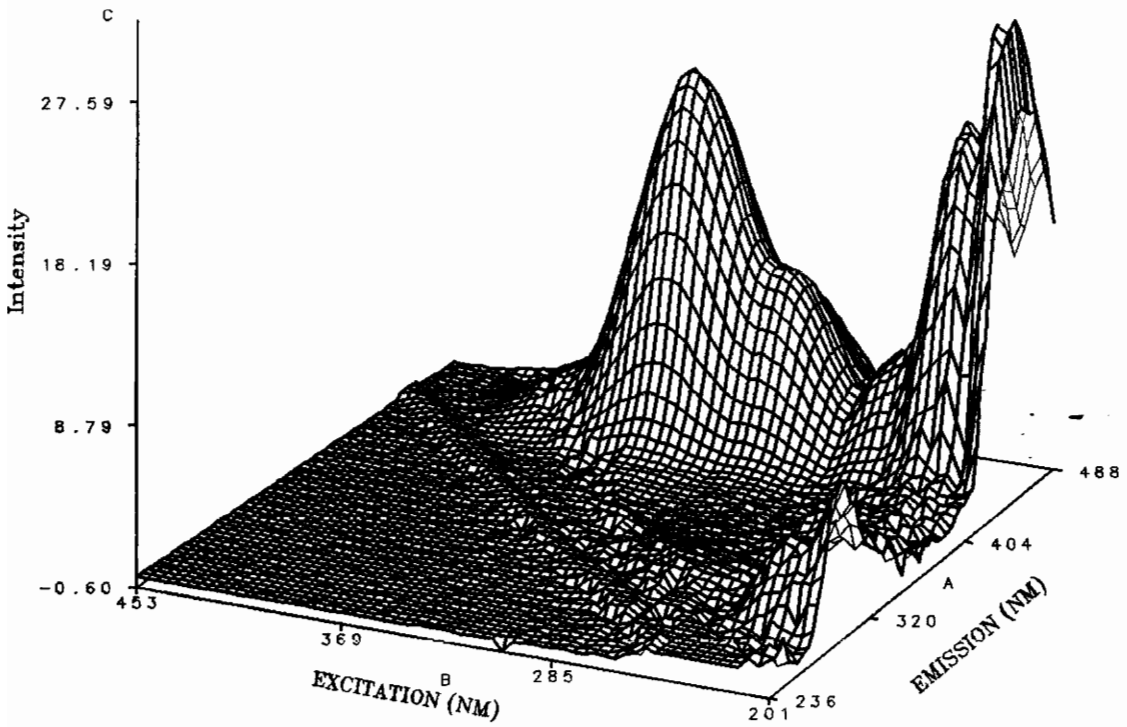


PPD. a) Structure
b) EEM

a)



b)



Esculin. a) Structure
b) EEM

VITA

George Asimopoulos was born in Kozani, Greece on September 12, 1964. In 1971 he moved with his family to Thessaloniki, Greece, where he received most of his grade school education. He graduated from 2nd High School of Thessaloniki, Greece, in 1982. In November 1986, he graduated from University of Ioannina, Greece, where he received a Bachelor of Science Degree in Chemistry. In June 1987, he entered the graduate program in the Department of Chemistry at Virginia Polytechnic Institute and State University. During his stay at Virginia Tech he was elected President of the Council of International Student Organizations (CISO).