Kinetic Study of the Speciation of Copper(II) bound to Humic Acid.

Maria Bonifazi

A Thesis

in

The Department

of

Chemistry and Biochemistry

Presented in Partial Fulfillment of the Requirements for
the Degree of Master of Science at
Concordia University
Montreal, Quebec, Canada

January , 1991

ABSTRACT

Kinetic Study of the Speciation of Copper(II) bound to Humic Acid.

Maria Bonifazi

The distinguishable species present in a solution of Cu(II) equilibrated with a soil humic acid are identified by a kinetic method of analysis based on ligand displacement reactions with 3-propyl-5-hydroxy-5-(d-arabino-tetrahydroxybutyl) thiozolidine-2-thione (PHTTT). An approximate Laplace transform is used to assign the minimum number of components and approximate their associated concentrations and dissociation rate constants.

The initial parameters are refined by a non-linear multi-exponential least squares regression routine. The two rate constants 0.093 and 0.0077 sec^{-1} consistently represent the middle and slow components of the copper humic acid complex (CuHA). The fast component is the major component present. Species concentrations vary in a reasonable way with pH and HA/Cu(II) ratio, supporting the conclusion that the speciation model has chemical significance.

Acknowledgements

I wish to express my sincere gratitude to Dr. C.H. Langford for his guidance and supervision of this research.

I wish to thank Dr. Pant for his valuable comments of the thesis draft.

TABLE OF CONTENTS

LIST OF TABLES

Page

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

A: Statement of the Problem:

The chemical speciation (1-3) of aqueous copper(II) with humic substances was studied using a kinetic method of analysis. Ligand-exchange reactions of copper-humic complexes with a colorimetric ligand, 3-propyl-5-hydroxy-5-(d-arabino-tetrahydroxybutyl) thiazolidine-2-thione (PHTTT), were examined spectrophotometrically following the formation of a common product as a function of time. A multi-site binding model was employed to determine the kinetic dissociation constants of the copper humate complexes. These rate constants indicate the relative labilities of each component present in humic acid.

The dissociation kinetics of copper complexes is of importance as it relates to metal bioavailability and toxicity (4,5). Copper can be present as cupric ions or complexed to organic colloids, hydrous oxide colloids, clays, or sorbed on bottom sediments as precipitates. The percentage of copper present in each of these forms depends on the pH and other inorganic and organic substances present in the water (6-10). Most workers (11,12) have concluded that the toxicity of copper to aquatic organisms correlated well with free cupric ion activity rather than total concentrations of metal ion in solution. Toxicity decreased when water-soluble compounds of unknown composition but with soil-like characteristics were added to the solutions. Conversely, high concentrations of organic matter can cause deficiency in copper which is a component of active sites of a number of enzymes and respiratory pigments.

B: Humic Acid

Humic substances are ubiquitous in the soil and water environments being formed by the microbial degradation of biomass and the synthetic activity of microorganisms (13-15). According to Reuter and Perdue (16), about 60 to 80% of the dissolved organic carbon and particulate organic matter is comprised of humic substances. Leaching of soils by surface run-off is responsible for the movement of soil humic acids toward rivers and lakes. The dissolved organic matter content of productive lakes is several mg/l (17). The random reactions in the formation of humic acid are influenced by variations in climate, vegetation, topography, and the age of the parent material (13). None the less, there are similarities in the composition and the properties of humics from diverse sources.

Due to the importance of natural organic matter in the form of humic and fulvic acids, an understanding of its characteristics and trace metal interactions are important in developing realistic models of speciation.

Organic matter extracted from soil is fractionated on the basis of solubility characteristics. Humic acid is soluble in alkali, insoluble in acid; fulvic acid is soluble in alkali and acid; humin, is insoluble in alkali (18,19). Although humic acid is an extremely complex mixture of compounds it cannot be fractionated into simpler pure compound components. Fulvic and humic acids range in molecular weight from a few hundred to several thousand (20).

In a study by Schnitzer, (21) the elemental analysis of humic acid in various regions shows the C content ranging from 53.8 to 58.7%, the O content from 32.8 to 38.3%.

Percentages of H and N vary from 3.2 to 6.2 % and 0.8 to 4.3 %, respectively. The S content ranges from 0.1 to 1.5 %. The humic acid used in this study was Laurentian HA obtained from the soil of the Laurentian Forest Preserve of Laval University (22). The analytical properties are shown in Table l.

A variety of functional groups have been reported for humic acid (23-25). Most methods used for determining the types of ligands responsible for metal binding are based on the acidic properties. Each method bears advantages and disadvantages (26).

Humic acid is highly substituted with carboxyl gro'·ps and in aqueous solution it constitutes a weak acid type of polyelectrolyte. A popular method for determining the total acidity is addition of $Ba(OH)_2$ to humic acid followed by back titration with standard acid (27).

$$2\,HA + Ba(OH)_2 \longrightarrow BaA_2 + 2H_2O + \text{unused base} \qquad (1.1)$$

Potentiometric titration with sodium hydroxide gives two end points. The total carboxyl content can be found by calculating the Gran's functions for the particular case of a weak acid having two end points (28-29). The phenolic-OH is obtained by the difference of total acidity from the -COOH content.

Other techniques have been used, such as, an iodometric method and reaction with lithium aluminium hydride for determination of total acidity (30). The -COOH is determined by methylation and subsequent saponification of methyl esters (31) or decarboxylation of aromatic acid with quinoline (32). The method by Ubaldini has been

Table 1 Analytical Data for Laurentian HA (22).

Element	Content (%)	Element	Content (%)
C	51.9	Na	0.0006
O	39.9	K	5×10^{-5}
H	5.5	Ca	0.0004
N	2.3	Mg	not detected
S	0.26	Fe	1.6

Total Acidity	7.60 mmol/g (±10%)
COOH	2.50 mmol/g (±5%)
OH	5.10 mmol/g (±10%)
Ash	< 0.1 %

used to determine phenolic-OH (33).

The major propeties of humic acid are its polyfunctionality, polyelectrolytic character, and conformational variability. The humic acid conformation and structure is controlled by the sample concentration, the pH of the solution, and the ionic strength of the medium. In an acidic medium humic acid exists as a separate physical phase. At low ionic strength Donnan potentials developed between the inside and outside of the humic gel control the sorption of electrolytes. As the pH increases. the carboxyl functional groups deprotonate and disrupt the forces holding together the three dimensional network of the macro-gel (34). The nature of these forces is likely to be hydrogen bonds between electron donor groups and protons (35). As the external solution rises 4 or 5 pH units the humic acid consists of a polyelectrolyte and a cation exchange gel in contact with an aqueous phase. The molecule assumes a stretched configuration due to anion site electrostatic repulsion. The hydration also increases with an increase in the number of dissociated groups and dehydration can be an irreversible process. Marinsky described the system as a micro-gel system impermeable to electrolytes (36-38).

Metal ions bind to humic acid via many co-ordinating grcups. In order to obtain specific information on the mechanism of reaction of metal ions (Cu, Zn, Fe, Al) with organic matter, Schnitzer and Skinner (39) selectively blocked the functional groups and measured the decrease in metal retention. Blocking acidic carboxyls or phenolic hydroxyls caused significant reductions in metal retention. Both groups appeared to react simultaneously with the metal ions. Copper chelation with salicylic and phthalic unit groups are both possible with humic acid as shown in fig 1.

The quantity of Cu(II) bound by humic acid increases with an increase in pH, molecular weight fraction, functional group content and decreases with ionic strength (40-43). Beckwith (44) concluded that the order of stabilities of metal complexes with humic substances followed the order of the Irving-Willians Series Pb(II) > Cu(II) > Ni(II) > Co(II) >Zn(II) > Cd(II) > Fe(II) > Mn(II) > Mg(II).

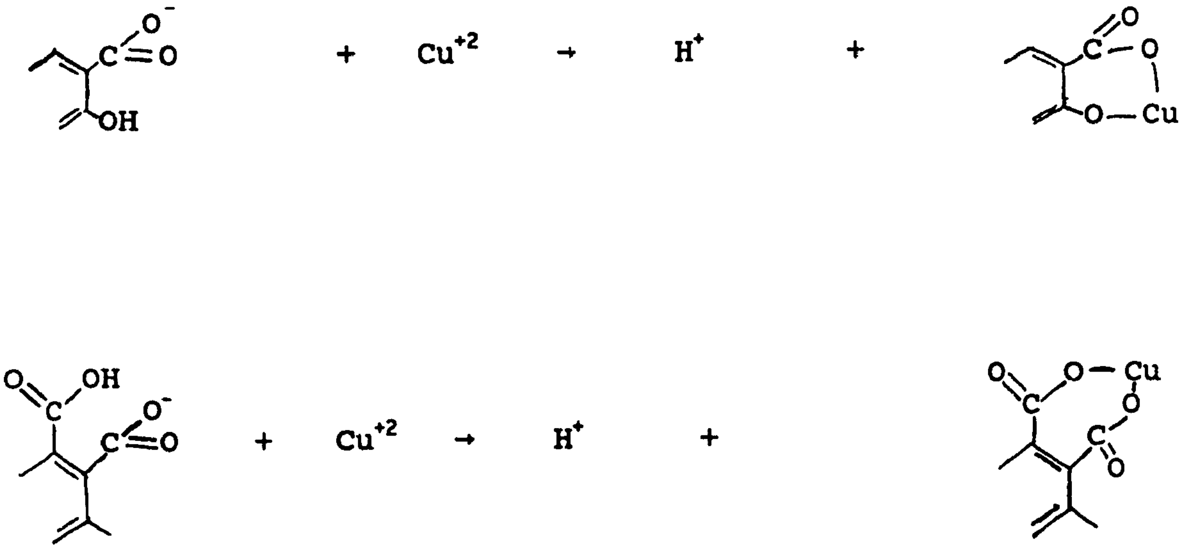


Figure 1: The copper chelation reactions with salicylic type and dicarboxylic type sites are possible for humic acid.

C: Strategy of approach

Dissociation of CuHA complexes has been observed via ligand exchange with PHTTT. The kinetic scheme is given below

$$CuHA_i \xrightarrow{slow} Cu^{+2} + HA_i \qquad (1.2)$$

$$Cu^{+2} + 3\ PHTTT \xrightarrow{fast} Cu(PHTTT)_3 \qquad (1.3)$$

where $CuHA_i$ is a kinetically distinguishable component. The color forming species reacts with all $CuHA_i$ species to form a common product for colorimetric monitoring. The product formed with PHTTT is stable and equation 1.3 has a known rate law. The rate determining step is the dissociation of the CuHA complexes. As well, with an excess of PHTTT equation 1.3 becomes pseudo-first order. Thus product formation may be expressed by equation 1.4

$$\frac{d[\ Cu(PHTTT)_3\ ]}{dt} = \sum_{i=1}^{n} k_i\ [\ CuHA_i\ ] \qquad (1.4)$$

where the k_i is the first-order dissociation constant for the i-th component. The integrated form of equation 1.4 gives the concentration of $Cu(PHTTT)_3$ as a function of time and is represented by equation 1.5

$$P(t) = \sum_{i=1}^{n} C(0,i)[\ 1 - \exp(-k_i\ t)\ ] + X \qquad (1.5)$$

where P(t) is the optical absorbance at any time t, C(0,i) is the initial value of the concentration of the i-th component expressed in units consistent with P. The X term includes the humic acid, the PHTTT solution, and reactions of copper ions with PHTTT in a time short compared to the time scale of the analytical method. These copper species include the hexaaquo ion and any weakly bound copper complexes since dissociation ratios of simple Cu(II) complexes are typically of the order of 10^6 sec^{-1} and greater.

The buffered PHTTT reagent controls kinetically important parameters such as pH and ionic strength ensuring reproducible rate constants. The kinetic values give a measure of the relative lability of the various components of CuHA before addition of PHTTT but they are not rate constants for the conditions of equilibration which defined the sample.

The number of components and their corresponding parameters k_i and C(0,i) must be extracted from the P(t) function. With both k_i and C(0,i) being unknown, nonlinear regression is required. However, an initial approximations of the values must be obtained beforehand. The multi-component kinetic analysis introduced by Olson and Shuman (45) identifies the minimum number of components with their corresponding rate constants and concentrations. The method employs an approximate Laplace transform for the time function P(t) and a kinetic spectrum is obtained. The number of components are seen as peaks. The application of this method will be discussed in Chapter 2.

Chapter 2: THEORY

A:Representative Models for Humic Acid

Equilibrium and kinetic models have been used to describe the CuHA complexes (46-49). Relatively labile systems can be modeled by equilibrium considerations whereas relatively inert systems require combined kinetic and equilibrium models to properly predict metal partitioning. Thermodynamic stability data alone address but one aspect of the biogeochemistry of metals. Another aspect, the kinetic dissociation of metal complexes as it relates to metal bioavailability and toxicity must also be considered. The central characteristic of CuHA equilibria is that copper ions are complexed to a mixture of heterogeneous sites and there is an electrostatic potential at each reaction site which varies with copper and proton loading. These various models must make assumptions concerning the number and nature of the reactions involved (50-55). It necessarily follows that the fit to the model proposed yields information on the degree and nature of heterogeneity.

The most appropriate means of rigorously representing complexation properties minimizes the assumptions made concerning the number and nature of the reactions involved. The heterogeneity and polyelectrolytic nature of the sites are an intrinsic property of the complexants. The model considers binding to occur at a large number of nonequivalent sites (56,57).

A binding model can be evaluated and a binding property illustrated by having the experimental data cover the widest possible range of pertinent physical/chemical conditions. Titration curves are most often used to test equilibrium models (58-60). A cupric ion selective electrode can measure the activity of the uncomplexed cation at distinct levels of total copper dissolved in the presence of humic acid (organic carbon). The heterogeneity of the mixture causes the complexing effect to be spread over a large titration range. For reasons of precesion and instrumental limitations, free cupric ions are measurable over a restricted range and one will never be able to observe all possible complexing sites using a single method. The stability constants extracted from the titration curve are valid within the experimental range.

The proposed approach to interpreting complexation requires accumulation of a large number of experimental data under a wide range of different conditions. The data is transformed into a function sufficiently normalized to allow relevant thermodynamic information and comparison of results. The potential usefulness of these models lies in minimizing the number of conditionality factors. The transformation of such normalized spectra will be discussed. Both the Laplace transform method proposed by Shuman (45) and the differential equilibrium function proposed by Gamble (61) have been used to model trace metal complexation (25,62). The discussion that follows shows how such normalized functions can be obtained.

In considering the equilibrium case of Cu(ll) complexing with a sample of HA, equation 2.1 is obtained by applying the law of m s action to the whole sample all at

$$Cu^{+2} + L \rightarrow CuL \qquad K = \frac{M_c}{M_{Cu}\, M_L} \qquad (2.1)$$

once where M_c is the concentration of metal humic complexes, M_{Cu} is the concentration of copper, and M_L is the ligand concentration. The equilibrium function K is dependent upon the composition of the chelating solution (25). The properties of K can be related to the properties of the components of the mixture of chelating sites. Each ligand in the mixture comes to its own equilibrium in its reaction with Cu(II). The i^{th} case is described by equation 2.2

$$Cu^{+2} + L_i \rightarrow CuL_i \qquad K_i = \frac{M_{ci}}{M_{Cu}\, M_{Li}} \qquad (2.2)$$

where M_{ci} is the concentration of the copper complexing with the i-th ligand, and M_{Li} is the concentration of the i-th ligand. In equation 2.3 the equilibrium function K is obtained by summing over all component ligands. From equation 2.2 one gets the rearranged form, equation 2.4

$$K = \frac{M_{c1} + M_{c2} + M_{c3} + \ldots + M_{cn}}{M_{Cu}\ (\ M_{L1} + M_{L2} + M_{L3} + \ldots + M_{Ln}} \qquad (2.3)$$

$$M_{Ci} \ = \ K_i\, M_{Cu}\, M_{Li} \qquad (2.4)$$

$$\Delta X_{Li} = \ M_{Li}/C_L \qquad (2.5)$$

Equation 2.5 describes the amount of free i-th ligand as a mole fraction, $\Delta X_{L,i}$, of the total amount of all the ligands, C_L. Substituting equation 2.4 into equation 2.3 for each copper complex component, and writing the free i^{th} ligand as a mole fraction form gives

$$K = \frac{\sum_{i=1}^{n} K_i X_{Li}}{X_L} \qquad (2.6)$$

$$X_L K = \int_0^L K_i \, dX_L \qquad (2.7)$$

By determining the average equilibrium function, K , one can get an insight into the intrinsic or differential equilibrium function, K_i. The average equilibrium function is a weighted average of the differential equilibrium function, K_i of all of the components. Because K approximates a continuous function, the summation of equation 2.6 can be replaced by an integration in equation 2.7. Rearranging equation 2.7 and differentiating gives an equation for K_i in terms of the experimental quantity K.

$$K_i = \left[\frac{d(K X_L)}{d X_L} \right]_{X=Xi} \qquad (2.8)$$

A plot of KX_L verses X_L , where X_L is the mole fraction of the ligand mixture that still remains free, enables the differential binding function and free energy to be recovered from the first derivative at the point X_{Li}. The value of K will be unique at any given point in a titration. It will depend upon the complexing site composition through the K_i and X_i terms. As the titration proceeds the degree of occupation of each site will vary. A significant advantage of K is its direct and simple computation.

An alternative treatment, called the Laplace transform method, was suggested by Shuman for equilibrium and kinetic data analysis (45,62). The normalized function is a frequency distribution function of site concentration as a function of the equilibrium constant or dissociation rate constant.

The alternative approach which Shuman developed for the equilibrium model represents the titration at each point by the binding parameter *V*. This is given by equation 2.9

$$V = \frac{[\mathrm{ML}_i]}{\mathrm{L}_T} = \frac{K\,\mathrm{M}_{Cu}}{1 + \mathrm{KM}_{Cu}} \qquad (2.9)$$

In the case of a continuous distribution of ligands equation 2.9 can be written as equation 2.10

$$V = \int_0^K \frac{X(\log \mathrm{K})\,\mathrm{K}\,\mathrm{M}_{Cu}}{1 + \mathrm{K}\,\mathrm{M}_{Cu}}\, d(\log \mathrm{K}) \qquad (2.10)$$

where the product *X*(log K)d(log K) is equivalent to the site type concentrations. A numerical approximation developed by Ferry (77) estimates *X*(log K) from the experimental K vs M_{Cu} function.

The derivation of the normalized function for a kinetic analysis is briefly described below.

The concentrations of all CuHA complexes at time t are given by:

$$\mathrm{C(t)} = \Sigma\, \mathrm{C(i)} \exp(-\mathrm{k_it}) \qquad (2.11)$$

The model (57,58) considers binding to occur at a large number of nonequivalent sites and k is treated as a variable. The integral form of equation 2.11 is given in equation 2.12

$$C(k,t) = \int_0^\infty F(k,t) \exp(-k,t)\, dk \qquad (2.12)$$

C(k,t) is the Laplace Transform of F(k,t). The approximate solution for F(k,t) is found by inverting the integral with the Post-Widder equation 2.13 (63,64,77).

$$F(k,t) = \lim[\ (-1)^m/m!\ ]\ (m/k)^{m+1}\ \frac{d^m C(m/k)}{d\ t^m} \qquad (2.13)$$

A solution can be obtained for the function H(k,t) by using the first and second derivatives of C(k,t) with respect to ln t.

$$H(k,t) = \frac{\partial^2\ C(k,t)}{\partial\ (\ln t)^2} - \frac{\partial\ C(k,t)}{\partial(\ln t)} \qquad (2.14)$$

H(k,t) is a distribution function, from which a spectrum is obtained by plotting H(k,t) verses ln t. The Laplace profile for a copper humic acid dissociation is shown in figure 4. The 'affinity spectrum' obtained gives a series of peaks with peak maxima located at abscissa values corresponding to the dissociation rate constants of the individual binding sites and peak areas equal to the initial concentration of Cu(ll) bound to these sites. The integer m, used to approximate the 'true' spectrum is given a value of two. Larger values result in numerical instability and lower values lose the resolution of the spectrum (65). The time t corresponds to 2/k.

To apply equation 2.13 to the concentration function, it must be smoothed to enable computation of the derivative. However, undersmoothing can cause distortion of shape and oversmoothing causes widening of the spectrum and the appearance of additional peaks. Numerous authors (65-67) have tested with data simulations the problems of resolving the spectrum with noisy data. Results are variable and artifacts are easily generated. Another method must be provided to support the results obtained from the kinetic spectrum. The species' concentration and the species' rate constants obtained from the kinetic spectrum were fitted with equation 1.5 using nonlinear regression (66-69). The appropriateness of the model was confirmed by obtaining reproducible binding parameters well fitted to equation 1.5. The change in solution conditions should cause a corresponding change in the parameters as predicted by the chemical properties of humic acid. The kinetic spectrum has been shown to provide useful estimate of the initial parameters required in the nonlinear regression. Unlike the nonlinear regression, the affinity spectrum has the advantage of requiring no prior knowledge of the rate constants, initial concentrations, or the number of components in humic acid. All that is needed is for the reactions to be first-order or pseudo-first order.

B: NONLINEAR REGRESSION

The nonlinear model is given by equation 1.5. The parameters enter the algebraic expression nonlinearly therefore an exact solution does not exist. The values of the parameters that best fit the data are obtained by the least squares criterion (LSQ) (70).

$$Q = \; [\, y(i) - Y(i) \,]^2 \qquad (2.14)$$

where y(i) are the experimental data points for the i-th point, Y(i) is the predicted value for y(i) obtained in equation(1.5) after substituting the initial parameters. The corrections to the parameters are calculated and the LSQ is calculated once again. The procedure is repeated until the least sum of squares is less than 10^{-8}.

The characteristics of the present nonlinear regression package are described by Mak and Langford (71-73). The popular gradient methods of Gauss and Marquardt are combined. More than one solution can exist and reliable estimates must be provided. A maximum of three components can be resolved. For a two component system the function Q defines a seven dimensional contour surface as the unknown parameters are varied. The minimum is found by a steepest descent type procedure This approach (72) determines, at each iteration, the stepsize and direction the function will decrease most rapidly.

The NLR can be readily computed for two and three components present in comparable quantities. As the rates for the components approach each other, the values of the initial estimates becomes crucial. The rate constants of components at comparable

comparable quantities. As the rates for the components approach each other, the values of the initial estimates becomes crucial. The rate constants of components at comparable concentrations are resolved when the k's are separated by a factor of 2 and a factor of 10 is required when concentrations differ (71). The regression package computes the exact values of all the partial derivatives that comprise the Hessian matrix N. It avoids convergence on negative values by using the coefficients of the fitted polynomial. The formation function of Cu(PHTTT)$_3$ was smoothed by a best fitted polynomial and equal time intervals were required to work in a minimum time span (70).

More than one solution can exist and reasonable first estimates need to be provided.

CHAPTER 3: EXPERIMENTAL SECTION

A:MATERIALS

All chemicals were reagent grade unless otherwise stated. The PHTTT was synthesized according to the reported (74) procedure. Potassium hydroxide pellets and concentrated nitric acid were obtained from Allied Chemicals. Bis(2-hydroxyethyl)imino-tris(hydroxymethyl)methane (Bis Tris) and copper sulfate pentahydrate were obtained from Aldrich Chemical Co. The extraction of humic acid from soil was carried out by Z.Wang of Concordia University according to the method described below (75,76,22).

The approximate value for the binding capacity of CuHA complexes was 2.5 mmoles per gram (22). This value was used to calculate the ratio of binding sites to copper atoms.

B: Synthesis of PHTTT

A mixture of 13.5 mL propylamine and 15 mL methanol was added to 27 gms of glucose. The mixture was heated at 60°C for 15 minutes. Hot ethanol (30 mL) was added and the mixture was left to stand overnight. To this mixture was added 60 ml ethanol and 21 mL carbon disulfide, refluxing below 60°C for 40 minutes. The solution was left overnight and the precipitate was filtered, and recrystallized twice from 70% ethanol 30% water solution. The white crystalline product reacted with aqueous copper(ll) ions to give a yellow solution.

The maximum absorbance of $Cu(PHTTT)_3$ at 434 nm had a molar absorptivity of 13,800 ± 500 at pH 6 and 0.1 M ionic strength.

C:Extraction and Purification of Laurentian Humic Acid

A 2 kg sample of air-dried soil was weighed into a polypropylene jug and extracted with 20 L of 0.5 M NaOH solution under N_2 atmosphere to avoid any oxidation of humic materials. The extraction was carried out for 24 hours. Any insoluble materials such as sand and clay were removed from soil extract by settling and centrifuging at 3500 rpm for 30 minutes.

A column (150 cm x 7.5 cm ID) filled with 4 kg of Dowex HCR cationic resin (reprocessed before use with 1N HCl and then washed completely with deionized water until the pH value of water was constant) was used to obtain HA. Care was taken not to allow air spaces between beads and to keep the liquid level above the bed top. The eluate flow rate was maintained at about 3-5 mL per minute. When the pH value of the eluate became greater than 7.0, the collection was stopped. This eluate was used to obtain the brown powder of fulvic acid (FA) by lyophilization technique .

The HA precipitated in the column was then washed out with 0.1 N NaOH. When the pH of the washed out HA solution became greater than 6.0, the collection was stopped. Approximately 11 L of eluate was collected. The eluate was then acidified with 6 N HCl to pH 1.5, and allowed to stand overnight. The supernatant was decanted and the resulting HA slurry was further washed with 0.5 M HCl and centrifuged at 3500 rpm for 30 minutes to remove remaining FA having smaller molecular wieght. Finally, it was suspended in deionized water and dialyzed using dialysis tubing of 1000 MW cut-off to remove the remaining HCl and excess salts. The external water was replaced every 3-5 hours for three times and then every 12 hours until no change in pH and no more chloride was detected. The absence of chloride ions were detected by treatment of the external solution with a solution of $AgNO_3$.

The dialyzed HA sample was then freeze-dried. A dark-brown powder was obtained which was stored in a dark brown container to prevent photochemical reactions. The analytical data of Laurentian HA is given in Table I.

D: Preparation of Equilibrated Solutions

The mole ratio of humic acid to Cu(II), defined in terms of humic acid functionality, was varied from 2:1 to 5:1 in unit steps at pH 6 and 7. Each solution contained $8.33 * 10^{-5}$ M $CuSO_4 \cdot 5H_2O$ and the appropriate proportion of humic acid. The solutions were adjusted to the appropriate pH with 0.001 M KOH or HNO_3. Solutions were equilibrated for 24 hours in darkness and their pH value confirmed one hour before the kinetic run. The PHTTT was present in approximately 30 fold excess to Cu(II), $7.49 * 10^{-3}$ M, buffered at pH 6 with 0.26 M BIS TRIS and adjusted with 0.1M HNO_3.

E:Kinetics:

Photometric measurements were performed on a Perkin-Elmer Model 552 spectrophotometer at 434 nm. The formation of $Cu(PHTTT)_3$ was initiated by injecting 1.5 mL of PHTTT solution into a cuvette containing 1.5 mL of the CuHA solution. The final molarities upon dilution had the values of $4.14 * 10^{-5}$ M $CuSO_4 \cdot 5H_2O$, $3.74 * 10^{-3}$ M PHTTT, 0.1 M BIS TRIS. The rate of dissociation was measured at constant pH and ionic strength. The excess reagent ensured pseudo-first order reaction. The injection required 1.5 seconds and reaction began at t=0 and was monitored till 500 seconds with an error of about 0.5 sec. In a few cases the initial points were truncated due to formation of bubbles which caused a large noise signal. The cell block was maintained at 24.5°C ± 0.5°C using a thermostated circulating water bath.

The background absorbance, the X term, includes the humic acid, the PHTTT solution, and reactions of copper ions with PHTTT in a time short compared to the time scale of the analytical method. These copper species include the hex-aquo ion and any weakly bound copper complexes since dissociation ratios of simple Cu(11) complexes are typically of the order of 10^6 sec^{-1} and greater.(73).

CHAPTER 4: Data Processing Technique

A: Reduction of experimental data

The mathematical model was studied efficiently with the ASYST software package whose graphics program and linear least squares polynomial fitting routine are built-in features. The algorithms are given in the appendix A.

Lavigne et al. (78) tested the validity of this model by simulating data generated according to equation 1.5. Noise was introduced by adding or subtracting up to 0.6% of the data using a random generator of the noise which was distributed normally. The noise level exceeded experimental noise. The values for the time of recordings as obtained from the actual experiments were used. The final values of k_i's and C_i's s obtained from the kinetic method were the parameters used to generate the data. In the simple 4 component system used to emulate the experimental data, the multi-component kinetic method appeared to be successful.

The Ni(II)-FA equilibrium was modeled with four components. However due to small differences in rate constants, (eg. 0.6 sec^{-1} and 0.2 sec^{-1}) peaks overlapped. A preliminary procedure was required in this case. The initial elapsed time during which the first component had essentially reacted completely with the colorimetric reagent was removed from the data. The slowest component was estimated from the plot of $\ln(A_{inf}-A_t)$ vs t by using the linear 'tail' section of the plot where only the slowest component contributes The contribution of this last exponential term is subtracted from the formation function P(t). The remaining two components were obtained directly from the

Laplace Transform. The parameters were finally fitted to the NLR.

This study uses a similar approach with two exceptions. The rate constant for the first component is not measurable in the time scale of mixing. The last component has a greater k value and is represented by a peak in the Laplace transform. Stripping this component causes overfitting of the data and irreproducible results.

The formation of the complex Cu(PHTTT)$_3$ with increasing time was measured spectrophotometrically at every 0.5 sec intervals from 0 to approximately 500 sec. Figure 2 is an example of the change in absorbance with time ‹f a typical kinetic run. The kinetic data is reduced in size from 1000 to 100 points while maintaining all pertinent information. The interval between successive points increases with time, resembling a logarithmic scale. The algorithm is called Standard.A as presented in appendix A.

The Laplace transform was obtained by smoothing the logarithmic form of the formation function with a best fit nth degree polynomial function. The logarithmic coordinates optimizes the fitting routine, giving greater weight to the initial points (58). Oversmoothing can generate small inflections in the formation function and cause additional peaks to appear in the affinity spectrum. Despite the practical limitations it still serves as a useful preliminary reference of the site distribution.

Figure 3 shows the natural log of the absorbance plotted against the natural log of time for a representative data set. The data was smoothed with a 7-th degree polynomial. The Laplace Transform was obtained by applying equation 2.14 to the formation function, P(m,t) which can be related to C(m,t) by a constant factor.

The area of a triangle was used to approximate the area under the peak. The base width used in the calculation was the value defined by Olson and Shuman to be 1.6973 at 50% of the peak height (77).

The fast component which includes the free cupric ion and the weakly bound Cu(II) with a dissociation rate greater than 0.093 sec^{-1} was obtained from the X-term in the nonlinear regression after subtracting the absorbance contribution of humic acid and colorimetric reagent. The X-term, the rate constants and concentrations of the middle and last component were fitted to the P(k,t) function by a non-linear regression routine. In the previous multi-component kinetic study of FeFA and AlFA complexes the initial estimates for the NLR were found using the Guggenheim method (79,80). This graphical approach consists of converting the originally observed data into two equal sets such that the absorbance at time t in the first set is paired with the absorbance at time t+Δt in the second set; Δ being a fixed time interval. The absorbance at time t and t+Δt are subtracted and the logarithmic form of the pseudo first order kinetics equation gives equation 4.1

$$\ln(A_{t+\Delta t} - A_t) = -kt + \ln(A_{inf} - A_o)(1-e^{-k\Delta t}) \qquad (4.1)$$

A plot of $\ln(A_{t+\Delta t} - A_t)$ against t gives a curvilinear plot. If the components are kinetically well separated the curve is comprised of distinguishable linear segments that give the value of k by the slope. Success depends on the nature of the observed data. The Laplace method is significantly less subjective.

Chapter 5: Experimental Results and Discussion

A Analysis of Data

The concentrations of the copper are above the range present in natural waters which has been estimated at 0.0006 to 0.4 mg/l (81). Even upon the addition of reagent solution the conditions are in between fresh and marine waters with a pH of 6-7 and ionic strength of 0.1 M. Samples are as near natural water conditions as the method permits.

The optimal analytical conditions inhibited the metal concentration from being lower than $4*10^{-5}$ M. The colorimetric reagent, 4-(2-pyridylazo)resorcinol (PAR), would increase the detection limit with its greater molar absorptivity (82) but it is not specific to copper (83) and consequently would also complex with traces of iron present in the humic acid.

The formation of $Cu(PHTTT)_3$ with time is shown in Fig 2 as a representative experimental run. With an absorbance of 0.86 and a background contribution of 0.30 the total contribution of the complex $Cu(PHTTT)_3$ is 0.56. Dividing this absorbance by the molar absorptivity will give the total kinetic copper added to the solution ($4.12*10^{-5}$ M). All the copper reacted in the time scale of the experiment. The change in absorbance from time 0 to 500 seconds is 0.09 indicating approximately 84% of copper humic complexes are very labile and dissociate in times less than 1 second which is the time required for mixing the reagent and CuHA solutions.

The rate of the hexaaquo ion with the colorimetric ligand is determined by the rate of expulsion of the coordinated water molecule. However the hexaaquo copper exchange rate is very rapid relative to other metal ions of toxic importance. The mechanism of the

$Cu(H_2O)_6^{+2}$ reaction with PHTTT (84,85) is shown below:

$$Cu(H_2O)_6^{2+} + PHTTT \overset{K}{\rightleftharpoons} \{Cu(H_2O)_6^{2+}.PHTTT\} \qquad (5.1)$$

$$\{Cu(H_2O)_6^{2+}.PHTTT\} \underset{k_{-2}}{\overset{k_2}{\rightleftharpoons}} Cu(H_2O)_5PHTTT + H_2\underline{O} \qquad (5.2)$$

Equation 5.1 has an equilibrium constant for ion pairing of approximately 1. The formation of $Cu(PHTTT)_3$ is limited by equation 5.2 and the rate law shown below has a pseudo first order rate constant of approximately 10^{-6} sec^{-1}

$$V = k_1\ [M] \qquad (5.3)$$

where k_1 is $k_2K[L]$.

Lavigne (66) proposed the present kinetic method and studied nickel speciation of fulvic acid. The water exchange rate was much lower and free aqueous nickel was kinetically measured. Having all the components measurable within the experimental time range simplified the task of validating the model. The copper speciation is however of greater environmental interest.

The representative Laplace transform in Figure 4 was obtained from the 7th degree polynomial fit to the log-log data in Figure 3. The H(k,t) distribution function in Fig 4 indicates two peaks for the middle and slow component in humic acid. Additional peaks can appear but they will fail to give reproduceable results or the NLR will not converge to the final solution for the parameters.

The formation of $Cu(PHTTT)_3$ with time at varying copper to humic acid ratio is shown in Figure 5 and Figure 6. The absorbance of humic acid and colorimetric reagent has been removed. The concentration of the stronger binding sites increases with

Figure 2 : Representative run of absorbance vs time for the experimental data at HA:Cu(II) ratio of 3:1 and a pH of 6.0.

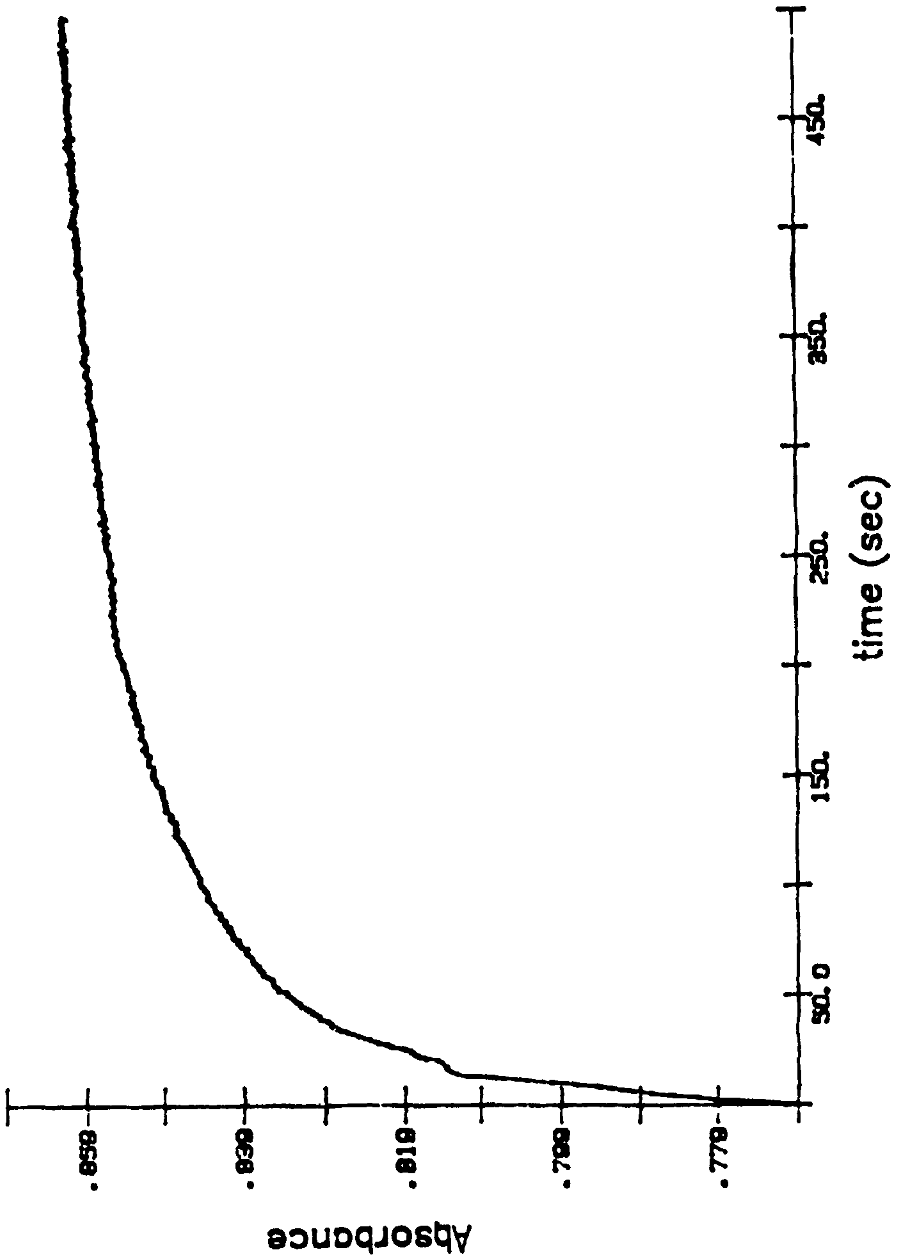
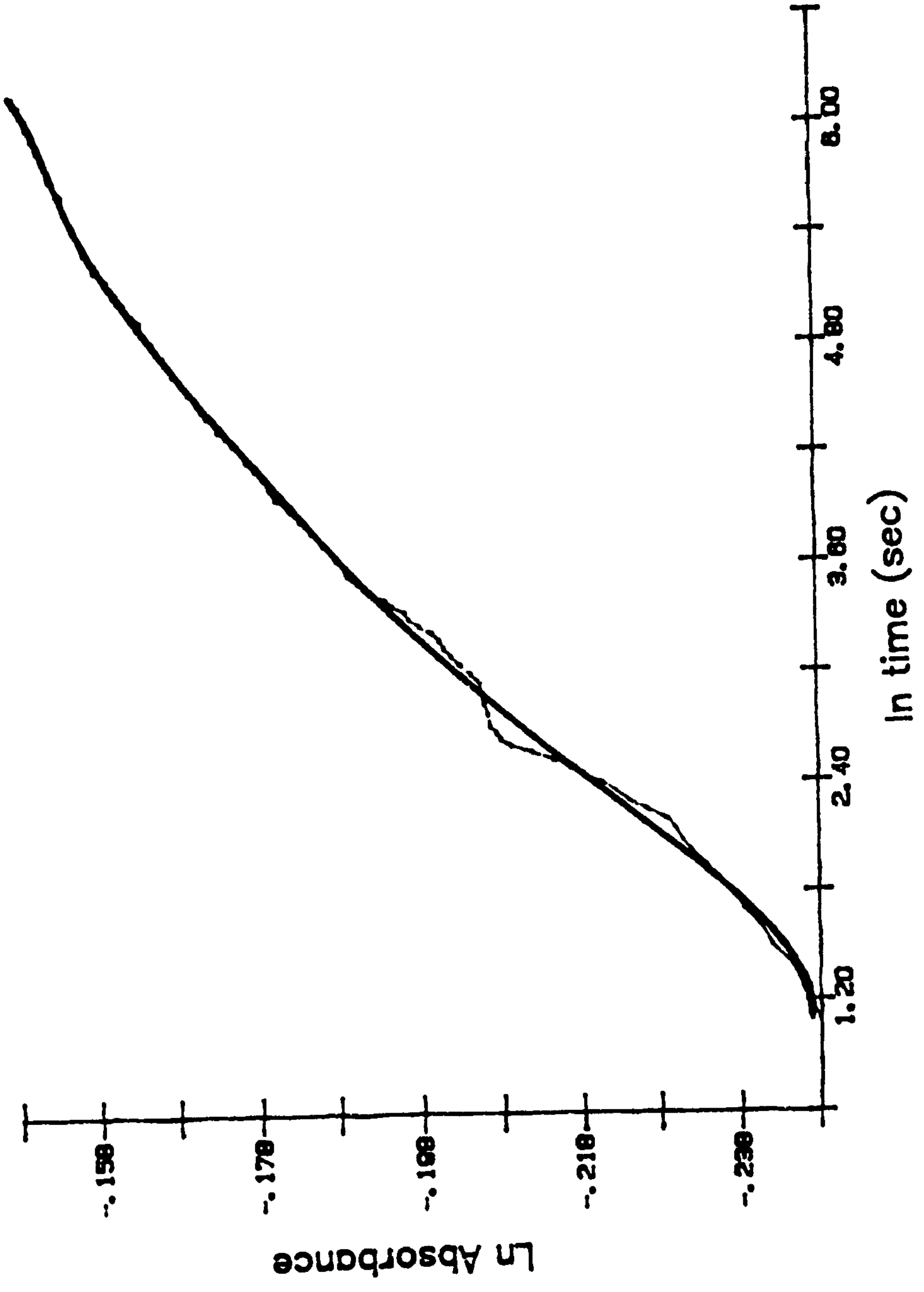


Figure 3: The log-log data of absorbance vs time of a representative gragh at pH of 6 and HA/Cu ratio of 3:1. The original data is plotted with a 7th degree least squares polynomial fit to the curve.



increasing humic acid to copper ratio.

The concentration profiles are given in fig 7 and 8 for a pH of 6.0 and fig 9 and 10 for a pH of 7.0. The concentrations of the $Cu(H_2O)_6^{+2}$ species and the fast component decrease with increasing HA:Cu(ll) ratio. The magnitude of this decrease is reflected in an increase in the middle and slow component. The mass action trends are shown to vary systematically with HA:Cu ratio.

The rate constants are given in Tables ll-V1. They are reproducible and stable with a standard deviation of 14 %. Individual binding sites can only be distinguishable if sufficiently different form one another. The heterogeneity of humic acid causes this to be unlikely. The components found in this study can be considered averages within an interval. These values can be considered mean dissociation rates for each component class indicating the chemical significance of the classes. The trends in concentration and the stablility of the rate constants allow the three component model to reasonably interpret the CuHA equilibrium speciation.

The values of rate constants and concentrations at pH 6.0 and 7.0 are similar as shown in Tables ll-Vll. This is reflected in the characterization of humic substances by potentiometric titration. Evaluation of titration points and the calculation of acid dissociation equilibrium functions as a function of their degree of ionization reveal two points of interest here. At a pH value of 5 to 6 the carboxy groups ortho to phenolic-OH are essentially all ionized and a large portion of all the functional groups before the first equivalence point are ionized (22,28,86). In this study it can be concluded that a one pH unit increase from 6.0 to 7.0 does not cause an appreciable change in the number of

Figure 4: The Laplace profile for the middle and slow component for a representative run at a pH of 6.0 and a HA/Cu(ll) ratio of 3:1. H(k,t), and C, values are expressed in terms of absorbancy changes of the kinetic reaction. Rate constants are in sec^{-1}.

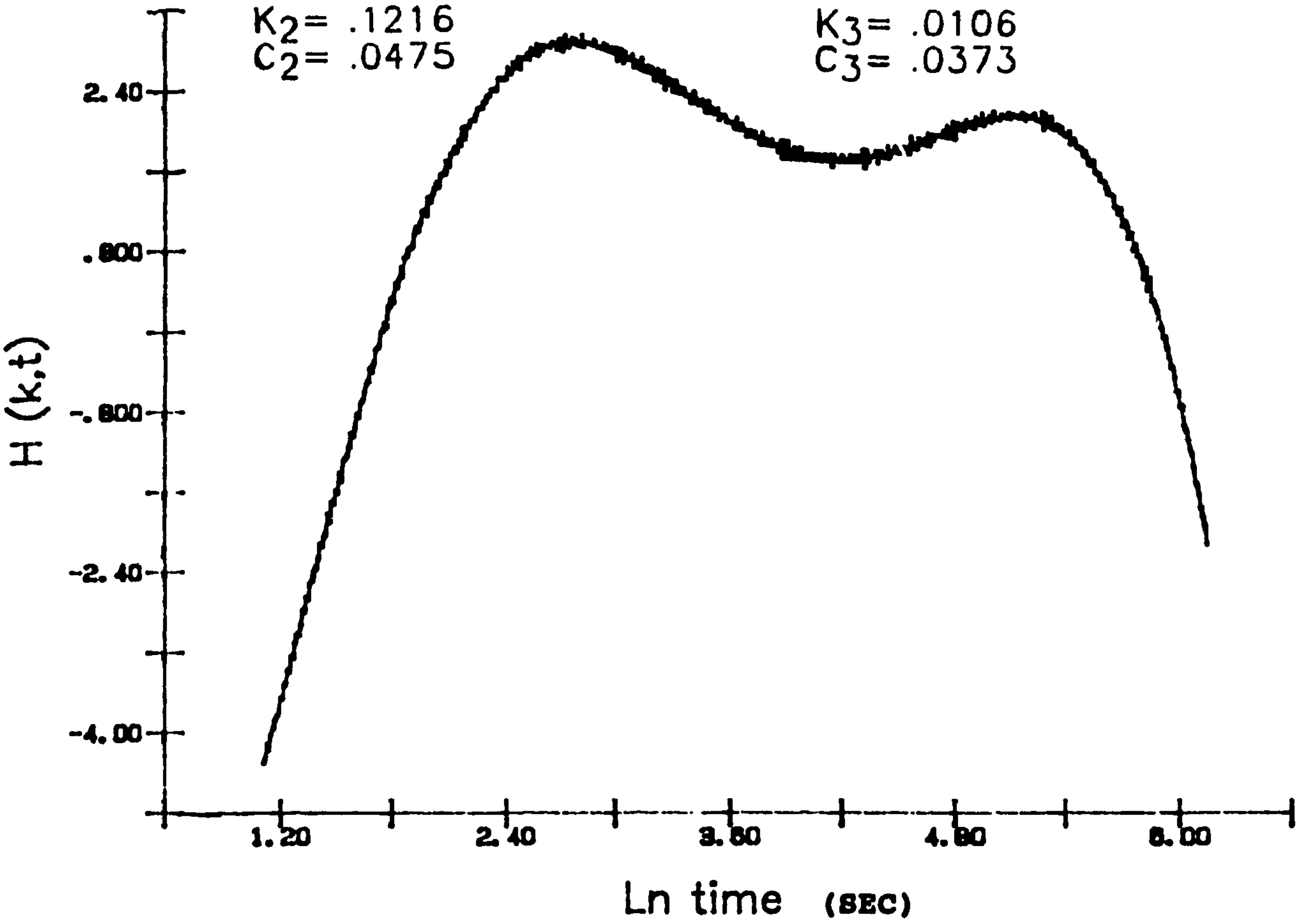


Figure 5: Representative experimental data, with the humic acid and colorimetric absorbance removed, of absorbance vs time at pH of 6.0 and a HA/Cu ratio of 2:1 and 3:1.

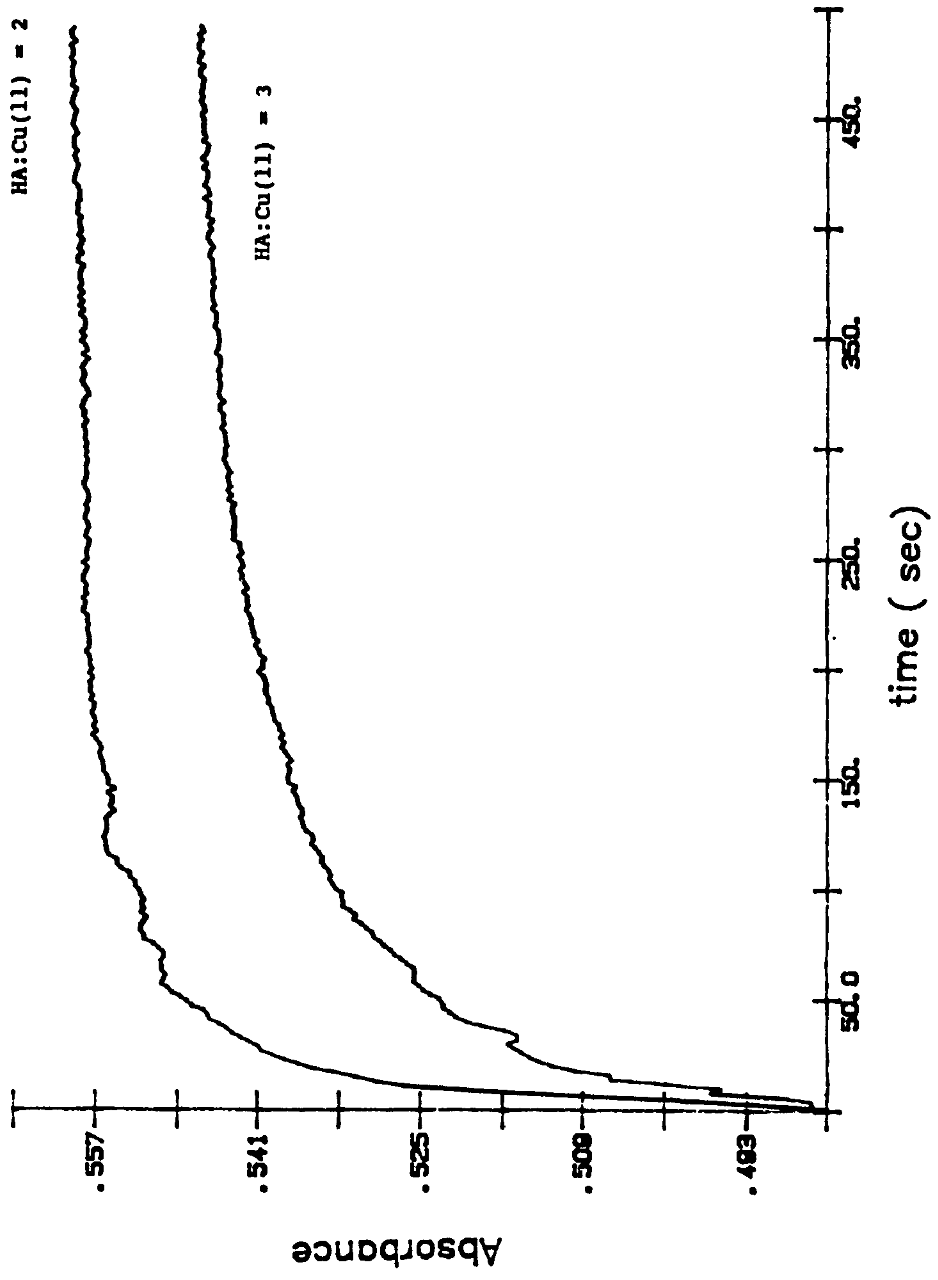


Figure 6: Representative experimental data, with the absorbance of humic acid and colorimetric reagent removed, of absorbance vs time for pH=6.0 and a HA/Cu ratio of 4:1 and 5:1.

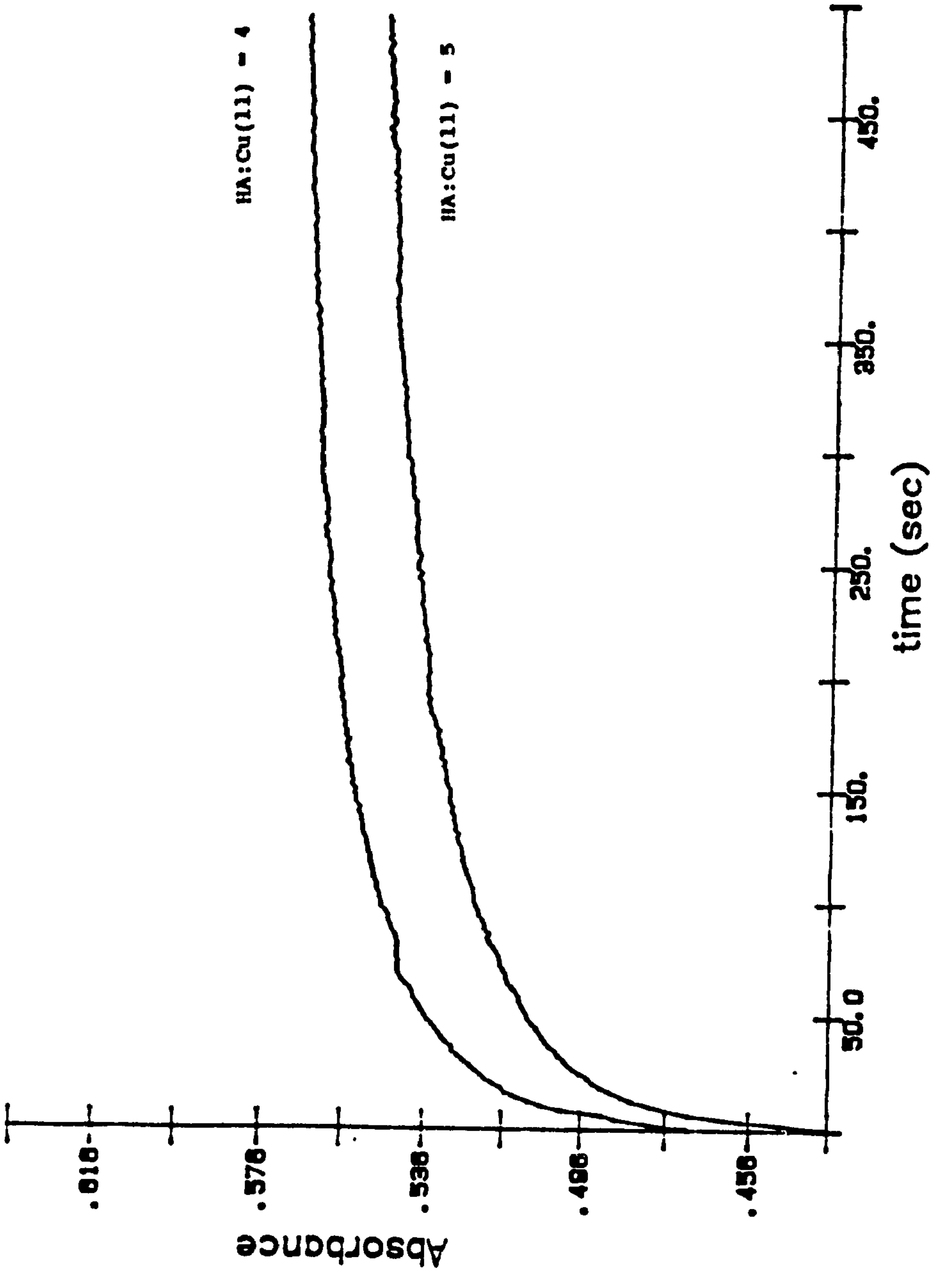


Figure 7: Profiles of the fast component concentrations at equilibrium of the three component model with HA:Cu(ll) ratios at a pH of 6.0. The kinetic concentration as a result of dilution with the PHTTT colorimetric solution are half these values.

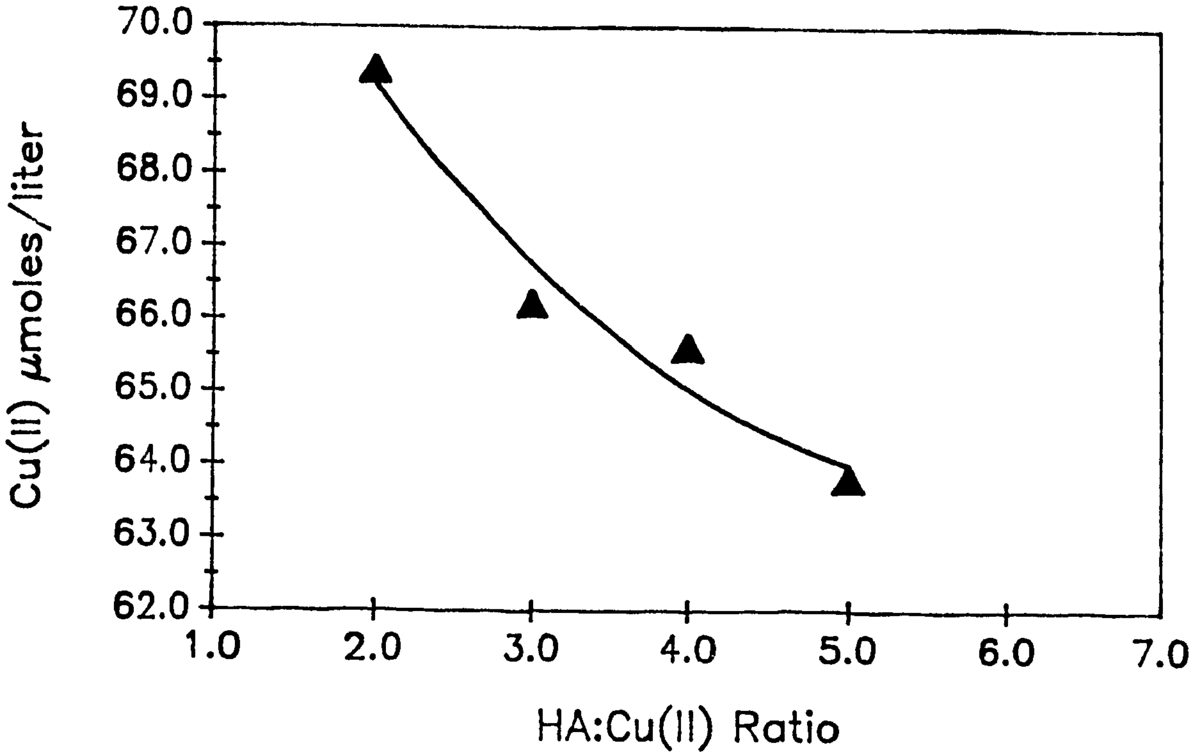


Table II: Normalized results for the kinetic concentration of the fast component for pH 6.0 and increasing HA:Cu(II) ratio.

HA:Cu(II)	CONCENTRATION[a]	
Ratio	Fast Component	Standard Deviation
2:1	34.7	2.76
3:1	33.1	2.57
4:1	32.8	1.26
5:1	31.9	1.62

[a] Concentration are in moles per liter X 10^{-6}.

Figure 8: Profiles of the middle and slow component concentrations at equilibrium of the three component model with respect to the HA:Cu(II) ratios at a pH of 6.0.

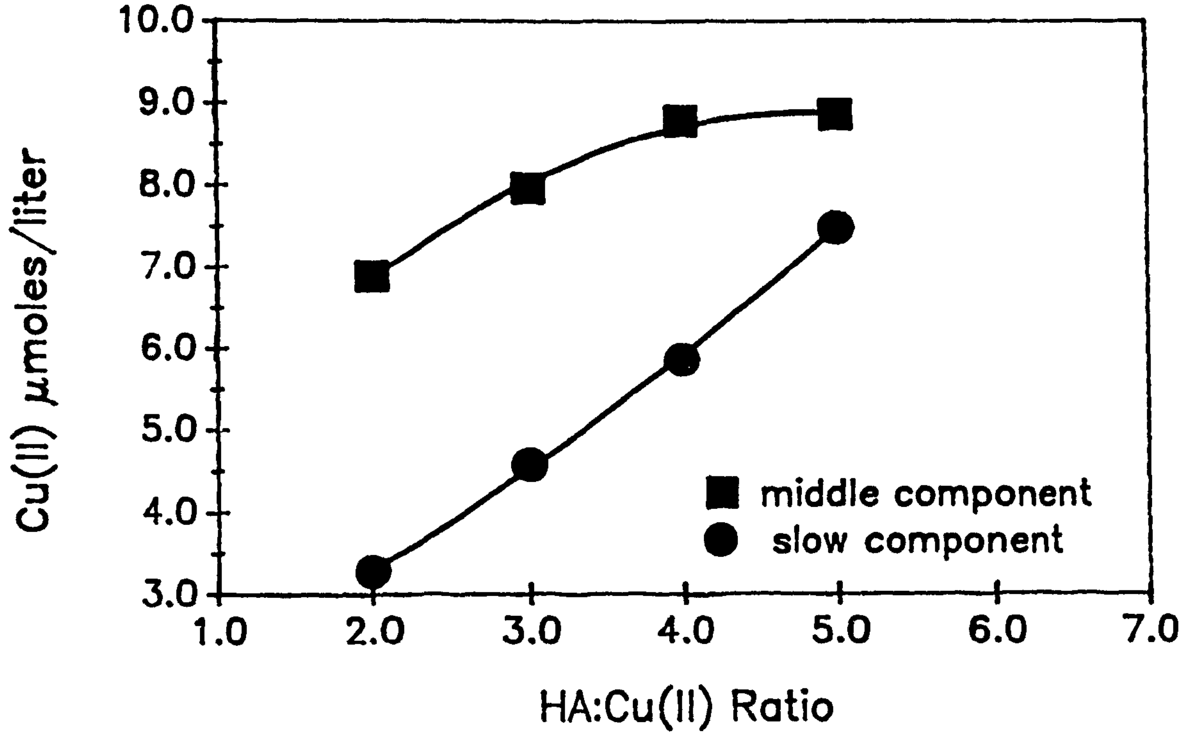


Table III: Normalized results for the kinetic concentrations of the middle component (C2) and thier corresponding rate constants obtained by NLR and estimated from the Laplace transform at pH 6.0.

HA:Cu	Rate Constants[a]			Concentrations[b]	
Ratio	Laplace	NLR	S.D.	C2	S.D.
2:1	0.176	0.12	0.022	3.44	0.993
3:1	0.159	0.095	0.023	3.98	0.977
4:1	0.142	0.081	0.019	4.39	0.704
5:1	0.129	0.091	0.035	4.43	0.836

[a] Rate constants are in sec^{-1}

[b] Concentrations of the middle component are in moles per liter x 10^{-6}.

Table IV: Normalized results for the kinetic concentrations of the slow component (C3) and the corresponding rate constants obtained by NLR and estimated from the Laplace transform at pH 6.0.

HA:Cu	Rate Constants[a]			Concentrations[b]	
Ratio	Laplace	NLR	S.D.	C.3	S.D.
2:1	0.015	0.0083	0.0026	1.64	0.54
3:1	0.016	0.0085	0.0015	2.29	0.46
4:1	0.019	0.0085	0.002	2.93	0.27
5:1	0.011	0.0071	0.001	3.74	0.20

[a] Rate constants are in sec^{-1}.

[b] Concentrations are in moles per liter x 10^{6}.

Figure 9: Profiles of the fast component concentrations at equilibrium for the three component model with respect to the HA:Cu ratio at a pH of 7.0. Concentrations after dilution with the colorimetric reajent are half these values.

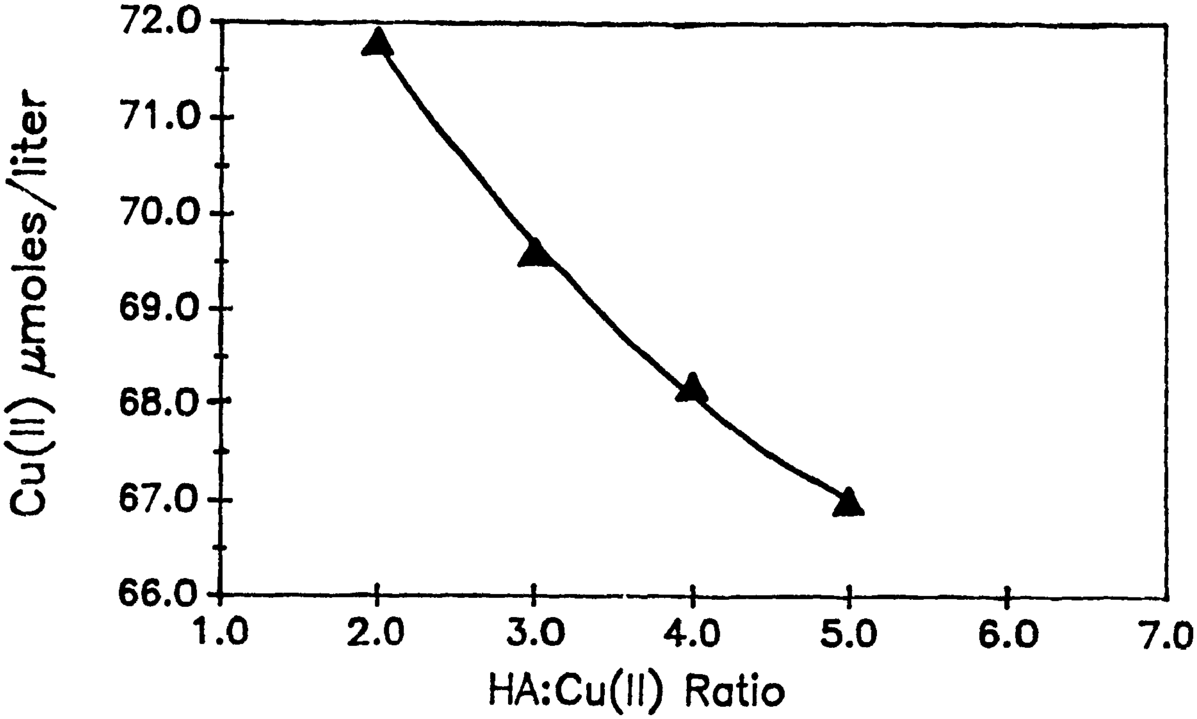


Table V: Normalized results for the kinetic concentrations of the fast component (C1) at pH of 7.0 and increasing HA:Cu(ll) ratio.

HA:Cu(II)		Concentration[a]
Ratio	C1	S.D.
2:1	35.9	4.11
3:1	34.8	2.58
4:1	34.1	1.25
5:1	33.5	0.94

[a] Concentrations are in moles per liter X 10^{-6}.

Figure 10: Profiles of middle and slow component concentrations for the three component model with respect to the HA:Cu(II) ratio at a pH of 7.0.

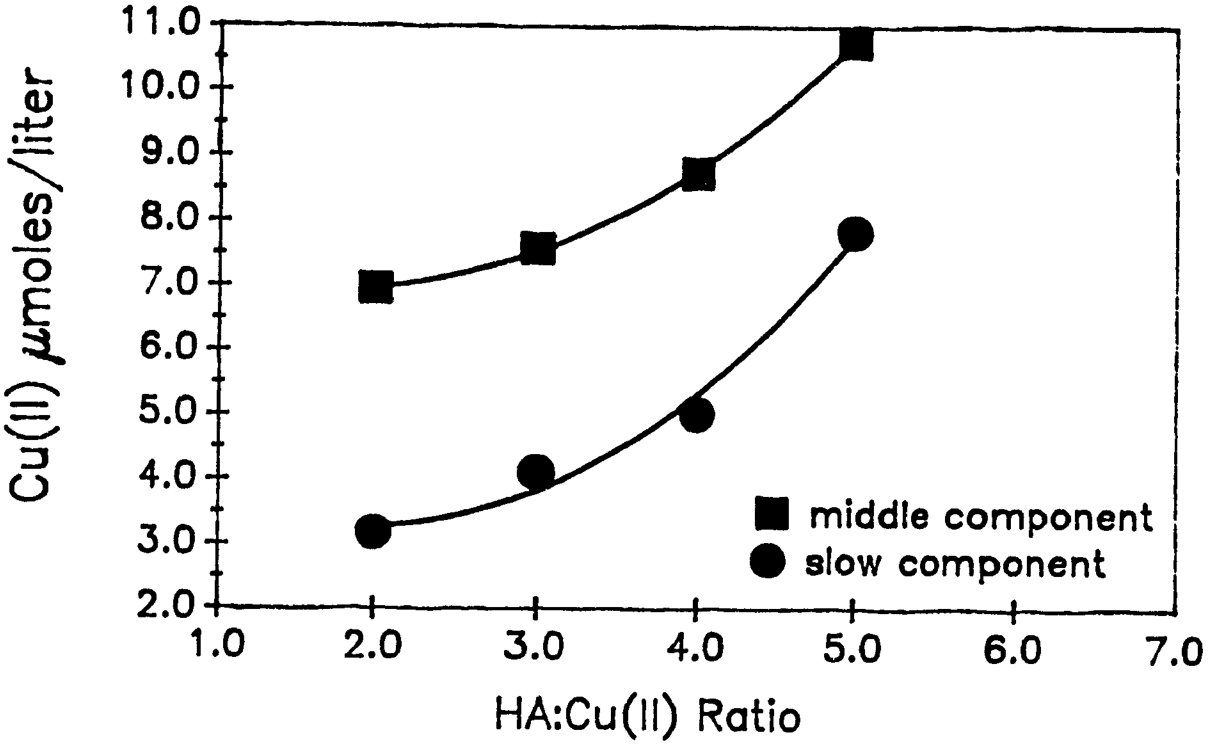


Table VI: Normalized results for the kinetic concentrations of the middle component (C2) and their corresponding rate constants obtained by NLR and estimated form the Laplace transform at pH 7.0.

HA:Cu	Rate Constants[a]			Concentrations[b]	
Ratio	Laplace	NLR	S.D.	C2	S.D.
2:1	0.15	0.10	.056	3.48	0.90
3:1	0.19	0.10	.033	3.79	0.53
4:1	0.15	0.083	.019	4.37	0.88
5:1	0.17	0.077	.046	5.38	0.54

[a] Rate Constants are in sec^{-1}.

[b] Concentrations are in moles per liter x 10^{-6} .

Table VII: Normalized results for the kinetic concentrations of the slow component (C3) and the corresponding rate constants obtained by NLR and estimated from the Laplace transform at pH 7.0.

HA:Cu	Rate Constants[a]			Concentrations[b]	
Ratio	Laplace	NLR	S.D.	C3	S.D
2:1	0.015	0.0083	.0033	1.59	0.23
3:1	0.019	0.0088	.0076	2.06	0.76
4:1	0.017	0.0066	.0015	2.51	0.85
5:1	0.020	0.0066	0.00093	3.92	0.93

[a] Rate constants are in sec^{-1}.

[b] Concentrations are in moles per liter x 10^{-6}.

dissociated acidic sites and this is reflected in the copper humic speciation.

B: Comparison with previous work

The dissociation reactions of colloidal complexes of Fe(lll) (87), Al(lll) (88), and Ni(ll) (78) have been previously examined by the sensitive spectrophotometric procedure employed in this work. As was previously mentioned the initial estimates of Fe(lll) and Al(lll) fulvic acid complexes (FeFA and AlFA) were determined by the Guggenhium method (80).

Iron speciation in aquatic systems include primarily hydrous oxide colloids and organic complexes (89). With limited solubility of monomeric Fe(lll) the spectrophotometric monitoring was obtained by reducing iron with hydroxylamine hydrochloride and the colorimetric reagent , ferrozine, binds to reduced Fe(ll). Non-linear regression yielded a major and minor component with both components found to react faster than the hydrous oxides. Although these complexes are more labile, filtration studies and light scattering experiments show there to be extensive aggregation. Fulvic acid also acted as a reducing agent in the absence of hydroxylamine hydrochloride. These features show the complex heterogeneous mixture of organic colloids. The rate constants are stable and the components identified appear to be chemically significant.

The speciation of AlFA complexes was shown to be more complex. However concentrations of aluminum ions and the pH of the medium affects hydrolytic precipitation in a complex manner even in the absence of organic compounds (90,91).

The AlFA equilibrium complexes were distributed in 5 classes with three distinguishable relatively labile forms of AlFA plus a slow but reactive component and an entirely unreactive and unrecoverable fraction of Al(lll). Small changes in the ratio of Al(lll) to FA leads to a change in the values of the rate constants and a complicated pattern of concentration distribution. The substantial fraction of non-labile Al(lll) attributable to the precipitation of hydrous aluminum oxides is a complex function of pH and concentration. In contrast to the Fe case, it is likely that the rate constants are nothing more than numerical parameters permitting reconstruction of the absorbance-time curve. The mixture is too complex to yield a chemically significant model partitioning into a small number of components.

The four components required to model the NiFA complexation follows a simpler pattern. The component rate constants k_1 varied randomly by 25% which was greater than the 12% deviation obtained with synthetic data at 0.6% noise. This deviation is attributed to the heterogeneity of humic acid. The component concentrations varied reasonably with pH and Ni:FA ratios.

This physical model agrees with the work done by Cabaniss (92). Although Lavigne et al. (78) worked at higher salt concentrations the data was fitted with a four component model using the same multi-kinetic method. The range of dissociation constants were similar although the lower average k_n values may be due to a difference in ionic strength or variability in the fulvic acid samples.

There is a difference in the objectivity of these two studies. In this work, as in Lavigne et al, the observed kinetics in the constant analytical solution conditions are related theoretically to the speciation of the original sample conditions. Cabannis (92) examined the effects of salt concentration, pH and Ni:FA ratio on the dissociation rates of NiFA complexes. The analytical solution conditions, with the exception of the colorimetric reagent, are identical to the sample solution conditions. This work varied the solution conditions for the dissociation reaction and produced variable rate constants for the components. These observed changes in the rate constants determine which variables affect kinetic modeling.

Olson and Shuman recently used photometric methods to study the kinetics of Cu-organic binding in an estuary (93). Most copper samples were adjusted to 28.4 uM Cu(II) and 84 mg dissolved organic carbon (DOC L^{-1}). The formation of $Cu(PAR)_2$ at a pH of 7 and 0.1 M ionic strength was monitored from 0 to 1835 sec.

The Laplace transform was used to ana'yse the data. The distribution of bound Cu(ll) was similar to our work. Using only the kinetic spectrum model however is dangerous. A number of authors have indicated problems with mathematical artifacts and another model should by employed to support the distribution.

Conclusions

The ability to measure and interpret metal speciation is limited by the ill-defined and complex qualities of humic acid.

In this study, equilibrated solutions of Cu^{+2} and Laurentian soil derived humic acid were treated with an excess of PHTTT. A multi-component kinetic approach was found capable of representing the rate of $Cu(PHTTT)_3$ formation.

The Laplace transform method derived by Olson and Shuman gave the minimum number of components and their corresponding rate constants and concentrations. These estimates were refined by the non-linear multi-exponential least squares regression routine.

The rate constants for the middle and slow component were 0.093 and 0.0077 sec^{-1}. The study varied the solution condition with an HA/Cu(II) ratio of 2 to 5 for a pH of 6.0 and 7.0. The copper equilibrium concentration was held constant at $8.33 * 10^{-5}$ M Cu(II) for all conditions.

As the concentrations of HA increased, the middle and slow component increase and the fast component decreased as predicted by the mass balance action. All the copper reacted with PHTTT at the end of 500 seconds suggesting all copper humic complexes are kinetically labile.

The rate constants were stable within a standard deviation of 14%. The concentrations varied systematically with HA:Cu(II) ratio. This led to the conclusion of having a valid 3-component model represent the data in a chemically significant way.

REFERENCES

1. Florence, T.M.; Batley, G.E.; CRC Reviews in Analytical Chemistry. 1980, vol.9, no.3, pp. 229-237.

2. Lund, W.; Fresenius Journal of Analytical Chemistry, 1990, 337, pp. 557-564.

3. Guy, R.D.; Chakrabarti, C.L.; Proceedings of the International Conference of Heavy Metals in the Environment, vol. 1; Hutchison, T., Ed., University of Toronto, Ontario, 1975, pp. 275-294.

4. Stevenson, F.J.; Biol. Sci. 1972, 22, p. 643.

5. Sunda, W.; Guillard, R.R.L.; J. Mar. Res. 1976, 34, pp.511-529.

6. Florence, T.M.; Batley, G.E.; Talanta, 1977, 24, pp.151-158.

7. Buffle, J.; Complexation Reactions in Aquatic Systems: An Analytical Approach, Ellis Horwood, Chichester. 1988

8. Turner, D.R.; Whitfield, M.; Dickson, A.G.; Geochim. Cosmochim. Acta. 1981, 45, pp. 855-881.

9. Benjamin, M.M.; Leckie, J.O.; J. Colloid Interface Sci. 1981, 83, pp. 410-419.

10. Lion, L.W.; Altmann, R.S.; Leckie, J.O.; Environ. Sci. Technol. 1982, 16, pp.660-666.

11. Bryan, G.W.; Proc. Roy. Soc. London B, 1971, 177, p. 389.

12. Nielson, E.S.; Wium-Andersen, W.; Marine Biology, 1970, 6, p.93.

13. Stevenson, F.J.; Humus Chemistry, John Wiley & Sons, New York, 1982, .

14. Swallow, K.C.; Westall, J.C.; McKnight, D.M.; Morel, N.M.L.; Morel, R.M.M. Limnol. Oceanogr. 1978, 23, pp. 538-542.

15. McKnight, D.M.; Morel, F.M.M.; Limnol. Oceanogr. 1979, 24, pp. 823-837.

16. Reuter, J.H.; Perdue, E.M.; Geochim. Cosmochem. Acta. 1977, 41, pp.325-334.

17. Sposito, G.; CRC Critical Review in Environmental Control, 13, March 1982.

18. Lee, Y.S.; Bartlett, R.J.; Soil Sci. Soc. Am. J. 1976, 40, 18, pp. 876-879.

19. Weber, J.H.; Wilson, S.A.; Water Res. 1975, 9, pp. 1079-1084.

20. Schnitzer, M.; Khan, S.U.; Humic Substances in the Environment, Marcel Dekker Inc., New York, 1972.

21. ref. 20), p.15

22. Wang, Z.; Ph.D. thesis, 1989, Concordia University, Montreal.

23. Schnitzer, M., Soil Organic Matter, Schnitzer M, and Khan S.U., Eds, Elsevier/North Holland, Amsterdam, 1978, 1.

24. Gamble, D.S., Can. J. Chem., 1970, 48, p. 2662.

25. Gamble, D.S.; Underdown, A.W.; Langford C.H., Anal Chem., 1980, 52, pp. 1901-1908.

26. ref. 13) pp. 234-239.

27. Schnitzer, M., Proc. Intern. Meetings on Humic Substances, 1972, Pudoc, Wageningen, pp. 293-310.

28. Gamble D.S., Can. J. Chem., 1972, 50, pp.2680-2690.

29. Gamble D.S., Can. J. Chem., 1970, 48, pp. 2262-2269.

30. Dubach, P.; Mehta, N.C.; Jakab, T.; Martin, B.; Boulet, N., Geochim. Cosmochim.

Acta, 1964, 28, p.1567.

31. Farmer, V.C.; Morrison, R.I.; Sci. Proc. Roy. Dublin Soc. Sec. A., 1960, 28, p.1567.

32. ref. 13) p.221

33. ref. 13) p.229.

34. Sojo L.E., Ph.D. thesis, 1988, Concordia University, Montreal.

35. ref. 7. Chapter 3.

36. Marinsky, J.A.; Ephraim, J.; Environ. Sci. Technol., 1986, 20, pp. 349-354.

37. Ephraim, J.; Alegret, S.; Matherthu, A; Bicking, M; Malcolm, R.L.; Marinsky, J.A.; Environ. Sci. Technol., 1986, 20, pp. 354-366.

38. Cabaniss, S.E.; Morel, F.M.M.; Environ. Sci. Technol., 1989, 23, pp. 746-747.

39. Schnitzer, M.; Skinner, S.I.M., Soil Sci., 1965, 99, pp. 278-283.

40. Ephraim, J; Marinsky, J.A., Environ. Sci. Technol., 1986, 20, pp.367-376.

41. Shuman, M.S.; Woodward, G.P.Jr., Environ. Sci. Technol., 1977, 11, pp. 809-813.

42. Herring, J.G.; Morel, F.M.M., Environ. Sci. Technol., 1988, 22, pp. 1234-1237.

43. Cabaniss, S.E.; Shuman, M.S.; Geochim. Cosmochim. Acta, 1988, 52, pp. 185-193.

44. Beckwith, R.S., Nature, 1959, 184, p.745.

45. Olson, D.L.; Shuman, M.S., Anal. Chem., 1983, 55, pp.1103-1107.

46. Pankow, J.F.; Morgan, J.J., Environ. Sci. Technol., 1981, 15, pp. 1306-1312.

47. Pankow, J.F.; Morgan, J.J., Environ. Sci. Technol., 1981, 15, pp. 1155-1164.

48. Fish, W.; Dzombak, D.A.; Morel, F.M.M.; Environ. Sci. Technol., 1986, 20,

pp.676-683.

49. Dzombak, D.A.; Fish, W.; Morel, F.M.M., Environ. Sci. Technol., 1986, 20, pp.669.

50. Ephraim, J.; Marinsky, J.A., Environ. Sci. Technol., 1986, 20, pp.367-375.

51. Buffle, J.; Greter, F.L.; Haerdi, w., Anal. Chem, 1977, 49, pp.216-222.

52. Wilson, S.E.; Kinney, P.; Limnol. Oceanogr., 1977, 22, pp. 281-289.

53. Perdue, E.M.; Lytle, C.R.; Environ. Sci. Technol., 1983, 17, pp. 654-660.

54. Shuman, M.S.; Cromer, J.L., Environ. Sci. Technol., 1979, 13, pp. 543.

55. Hunston, D.L., Anal. Biochem., 1975, 63, pp. 99-109.

56. Buffle, J.; Altmann, R.S.; Filella, M.; Tessier, A.; Geochim. Cosmochim. Acta, 1990, 54, pp. 1535-1553.

57. Buffle, J.; Altmann, R.S., Geochim. Cosmochim. Acta, 1988, 52, pp.1505-1519.

58. Cabaniss, S.E.; Shuman, M.S., Geochim. Cosmochim. Acta, 1988,52, pp.195-200.

59. Bresnahan, W.T.; Clarence, L.G.; Weber, J.H., Anal. Chem., 1978, 50, pp.1675.

60. Buffle, J.; Greter, F.L.; Haerdi, W., Anal. Chem., 1977, 49, pp.216-222.

61. Gamble, D.S.; Langford, C.H.; Environ. Sci. Technol., 1988, 22, pp. 1325-1335.

62. Shuman, M.S.; Collins, B.J.; Fitzgerald, P.J., and Olson, D.L., in " Aquatic and Terrestrial Humic Materials "; Christman, R.F.; Gjessing, E.T., Eds.; Ann Arbor Science, Ann Arbor, MI, 1983, pp. 349-370.

63. Widder, D.V.; " The Laplace Transform ", Princeton University Press, Princeton, N.J., 1946.

64. Bellman, R.; Kalaba, R.E.; Lockett, J.; " Numerical Inversion of the Laplace

Transform ", American Elsevier, N.Y., 1966.

65. Thakur, A.K.; Munson, P.J.; Humston, K.L.; Rodbard, D., Anal. Biochem., 1980, 103, pp. 240-254.

66. Lavigne, J.A.; Ph.D thesis, 1987, Concordia University, Montreal.

67. Truner, D.R.; Varney, M.S.; Whitfield, M.; Mantoura, R.F.C., Riley, J.P., Geochim. Cosmochim. Acta, 1986, 50, pp. 289-297.

68. Mak, M.K.S.; Langford, C.H., Comments Inorg. Chem., 1983, 2, pp. 127-143.

69. Langford, C.H.; Gamble, D.S., in " Complexation of Trace Metals in Natural Waters.", 1984, Martinus Nijhoff/Dr. W. Jung Publishers, The Hague, Netherlands, pp. 349-356.

70. Bard, Y., " Nonlinear Parameter Estimation. ", Academic, N.Y., 1974.

71. Mak, M.K.S.; Langford, C.H., Inorganica Chimica Acta, 1983, 70, pp. 237-246.

72. Marquardt, D.W; Chemical Engineering Progress, 1959, 55, pp.65-70.

73. Mak, M.K.S., Concordia University, Montreal, Quebec, Personal Communication.

74. Stiff, M.J.; Analyst, 1972, 97, p.146.

75. Wang, Z.; Pant, B.C.; Langford, C.H., Analytica Chimica Acta, 1990, 232, pp.43-49.

76 ref 13. Chapter ll.

77. Ferry, J.D., " Viscoelastic Properties of Polymers", 2nd ed.; Wiley: New York, 1963.

78. Lavigne, J.A.; Langford, C.H.; Mak, M.K., Anal. Chem., 1987, 59, pp. 2616-2620.

79. Kirkup, L.; Sutherland, J., Computers in Physics, 1988, pp.64-68.

80. Wood, R.E., "Introduction to Chemical Thermodynamics "; Meredith Corporation, New York, 1970, p. 101.

81. Dyressen, N; Plechanov, N.; Josefsson,B.; Lundquist, K., in " Aquatic and Terrestrial Humic Substances " ; Christman, R.F.; Gjessing, E.T., Eds.; Ann Arbor Science, Ann Arbor, MI, 1983, pp. 387-403.

82. Shibata, S., " Chelates in Analytical Chemistry ", Faschka, H.A., Barnard, A.J.,Marcel Decker Inc., New York, 1972, p. 131.

83. Yotsuyanagi, T.; Yamashita, R.; Aomora, K., Anal. Chem., 1972, 44, pp. 1091-1093.

84. Cotton, F.A.; Wilkinson, G., " Advanced Inorganic Chemistry", John Wiley & Sons Inc., N.Y.,1980, p. 1188.

85. Hoffmann, M.R.; American Chemical Society, 1981, 15, pp. 345-353.

86. Gamble, D.S., Can. J. Soil Sci., 1989, 69, pp.313-325.

87. Langford, C.H.; Wong, S.M.; Underdown, A.W., Can. J. Chem., 1981, 59, pp. 181-186.

88. Mak, M.K.S.; Langford, C.H., Can. J. Chem., 1982, 60, pp. 2023-2028.

89. Lefebvre, E.; Gegube, B., Wat. Res., 1990, 24, pp. 591-606.

90. Matijevic, E.; Jananer, E.F.; Kerker, M., J. Colloid Sci., 1964, 19, p. 333.

91. Matijevic, E.; Mathai, K.G., Hewill, R.H.O., Kerker, M., J. Phys. Chem., 1961, 65, p. 826.

92. Cabaniss, S.E., Environ. Sci. Technol. 1990, 24, pp. 583-588.

93 Olson, D.L.; Shuman, M.S., Geochim. Cosmochim. Acta , 1985, 49, pp. 1371-1375.

APPENDEX A: THE ASYST PROGRAMS FOR ESTIMATIONS

This is a listing of the programs that should be used to successfully find the parameters for the nonlinear regression. They are presented in the sequence that they are to be used.

The dimensions of array and scalar definitions are as follows:

```
>Under file DIM.ASY ;<

REAL DIM[ 2000 ] ARRAY Y
REAL DIM[ 2000 ] ARRAY X
>These arrays serve as the basic receiver/holders of the raw data as acquired from
DOS.<


REAL DIM[ 2000 ] ARRAY NEWY
REAL DIM[ 2000 ] ARRAY NEWX
>These arrays are for the smoothed/interpolated values for all newly generated x,y
fits.<


REAL DIM[ 2000 ] ARRAY DIFF1
REAL DIM[ 2000 ] ARRAY DIFF2
REAL DIM[ 2000 ] ARRAY NEWSET
>These arrays are used in the calculation of the Laplace profile.<


REAL DIM[ 2000 ] ARRAY LOGS
REAL DIM[ 2000 ] ARRAY SY
REAL DIM[ 2000 ] ARRAY SX
REAL DIM[ 2000 ] ARRAY BUFFER
>These arrays serve to make calculations without altering x,y, and newy<



>The scalars are listed below and can be better understood by studying the
algorithms.
```

```
INTEGER SCALAR NEXT
INTEGER SCALAR N.O.P
INTEGER SCALAR NUMBER
INTEGER SCALAR INT
INTEGER SCALAR SECT
INTEGER SCALAR COUNT
INTEGER SCALAR RATES
INTEGER SCALAR DEGREE
INTEGER SCALAR U
INTEGER SCALAR U1
INTEGER SCALAR U2
INTEGER SCALAR U3
INTEGER SCALAR U4
INTEGER SCALAR U5


>The following asyst words have been changed to reduce the amount of typing.
: G GRAPHICS.DISPLAY ;
: N NORMAL.DISPLAY ;
: C SCREEN.CLEAR ;
: ZZ STACK.CLEAR ;
: S STACK.DISPLAY ;
: A.P XY.AUTO.PLOT ;
: D.P XY.DATA.PLOT ;


>Under file "ODDS&.END " :<
>This file is used to set up the position on the sceen for the readout of the
arrays;<

: AR
array.readout
normal.coords
.6 .9 readout>position
world.coords
;


>To truncate the initial data set from time zero to time 'NEXT' ,use the word
TRUNK:<
: TRUNK
CR ." FROM WHICH INDEX ONWARD ? :" #INPUT NEXT :=
X SUB[ NEXT, N.O.P NEXT - ] X SUB[ 1 , N.O.P NEXT - ] :=
Y SUB[ NEXT, N.O.P NEXT - ] Y SUB[ 1 , N.O.P NEXT - ] :=
N.O.P NEXT - N.O.P := ;
```

>To obtain the concentration and rate constants in the "affinity spectrum, it is found by using the word AR followed by CC or KK respectively. Note that the values must be entered before using the words.<

```
: KK EXP 2. SWAP / . ;
: CC LABEL.SCALE.Y * 1.6973 * . ;
```

>Under file READ.ASY ;<
>To enter data from DOS use the word 'READ.BAS'. The number of pairs must be supplied.

```
: READ.BAS
CR. "ENTER THE FILENAME "
"INPUT DEFER> BASIC.OPEN
CR."ENTER THE EXPECTED NUMBER OF DATA PAIRS;"
#INPUT N.O.P :=
N.O.P 1 + 1 DO
BASIC.READ DROP
Y [ I ] :=
BASIC.READ DROP
X [ I ] :=
LOOP
BASIC.CLOSE
;
```

>Under file STAND.5;<
>Once the raw data has been entered, it must be reduced in size. To better apply the log-log polynomial fitting procedure the spacing between between pairs of points increases with time.

```
: STANDARD.A
X SUB[ 30 , 20 , 2 ] X SUB[ 30 , 20 ] :=
Y SUB[ 30 , 20 , 2 ] Y SUB[ 30 , 20 ] :=
X SUB[ 69 , 20 , 6 ] X SUB[ 50 , 20 ] :=
Y SUB[ 69 , 20 , 6 ] Y SUB[ 50 , 20 ] :=
X SUB[ 188 , 10 , 10 ] X SUB[ 70 , 10 ] :=
Y SUB[ 188 , 10 , 10 ] Y SUB[ 70 , 10 ] :=
X SUB[ 287 , 6 , 20 ] X SUB[ 80 , 6 ] :=
Y SUB[ 287 , 6 , 20 ] Y SUB[ 80 , 6 ] :=
X SUB[ 406 , 15 , 50 ] X SUB[ 86 , 15 ] :=
Y SUB[ 406 , 15 , 50 ] Y SUB[ 86 , 15 ] :=
X SUB[ 1155 , 8 , 50 ] X SUB[ 86 , 15 ] :=
Y SUB[ 1155 , 8 , 50 ] Y SUB[ 86 , 15 ] :=
300 1 DO
X [ I 1 + ] X [ I ] - X [ I 1 + ] X [ I ] - ABS / -1. =
IF I N.O.P := LEAVE THEN
LOOP
N.O.P OLD.N :=
X SUB[ 1 , N.O.P ] SX SUB[ 1 , N.O.P ] :=
Y SUB[ 1 , N.O.P ] SY SUB[ 1 , N.O.P ] :=
;
```

>File FIX.ASY ;<

The size of data points for this file is limited by the ASYST system. The limitations require that the size of the data to be fitted times the degree of polynomial fitting be less than 1000. This program gives a log-log plot of the experimental data. It than interpolates by logarithmic spacings the number of new pairs of points. The experimental curve is smoothed with by linear least squares regression having the new pair of points has x values. The user gives the choice of degree and the number of new pairs. It is suggested that for a full data set that a seventh degree polynomial fit be attempted first.

```
: FIX.XY
ZZ
CR ." # NEW PAIRS ? " #INPUT NUMBER :=
X SUB[ 1 , N.O.P ] LN
Y SUB[ 1 , N.O.P ] LN
A.P
CR ." DEGREE > " #INPUT DEGREE :=
X [ N.O.P ] LN X [ 1 ] LN - NUMBER 1 - / U :=
NUMBER 1 + 1 DO
I 1 - U * X [ I ] LN + EXP NEWX[ I ] :=
LOOP
X SUB[ 1 , N.O.P ] LN
Y SUB[ 1 , N.O.P ] LN
DEGREE LEASTSQ. POLY FIT
DUP COEFF SUB[ 1 , DEGREE 1 + ] :=
NEWX SUB[ 1 , NUMBER ] LN SWAP POLY [X]
NEWY SUB[ 1 , NUMBER ] :=
NEWX SUB[ 1 , NUMBER ] LN
NEWY SUB[ 1 , NUMBER ] DUP
EXP NEWY SUB[ 1 , NUMBER ] := LN D.P
;
```

>File LAP.ASY ;<

This is the file that calculates the Laplace transform to the smoothed data. After the affinity spectrum has been plotted, type AR to obtain cursur movement and type KK and CC to obtain the values for the maxima.<

```
: LAPLACE
NEWX SUB[ 1 , NUMBER ] LN NEWX SUB[ 1 , NUMBER ] :=
NUMBER 1 DO
NEWY [ I 1 + ] NEWY [ I ] - U / DIFF1 [ I ] :=
NEWX [ I 1 + ] NEWX [ I ] + 2. / NEWSET [ I ] :=
I NEXT :=
LOOP
NUBER 1 - 1 DO
DIFF1 [ I 1 +] DIFF1 [ I ] - U / DIFF2 [ I ] :=
NEWSET [ I 1 + ] NEWSET [ I ] + 2. / NEWSET[ I ] :=
LOOP
NUMBER 2 - 1 DO
DIFF1 [ I 1 + ] DIFF1 [ I ] + 2. / -1. *
DIFF2 SUB[ I ] + -1. * DIFF2 [ I ] :=
LOOP
NEWSET SUB[ 1 , NUMBER 3 - ]
DIFF2 SUB[ 1 , NUMBER 3 - ]
NEWX SUB[ 1 , NUMBER ] EXP NEWX SUB[ 1 , NUMBER ] :=
A.P
;
```

>File TRANS.FER :<

>This program gives a linear least square polynomial fit to the log-log data. The difference between this program and fix.xy is in the interval of the new data pairs. The new pairs in this program are spaced at equal time intervals in order to be compatible with the nonlinear regression. The user gives the choice of degree and the number of new pairs.

```
: NEW.XY
```

```
N CR ." NUMBER OF "
CR ." NEW PAIRS?" CR
#INPUT NUMBER :=
X SUB[ 1 , N.O.P ] LN
Y SUB[ 1 , N.O.P ] LN
CR ." DEGREE?" #INPUT DEGREE :=
DEGREE LEASTSQ.POLY.FIT
COEFF SUB[ 1 , DEGREE 1 + ] :=
X [ N.O.P ] X [ 1 ] - NUMBER 1 - / U1 :=
NUMBER 1 + 1 DO
I 1 - U1 * X [ 1 ] + NEWSET [ I ] :=
LOOP
NEWSET SUB[ 1 , NUMBER ]
LN
COEFF SUB[ 1 , DEGREE 1 + ] POLY[X]
LOGS SUB[ 1 , NUMBER ] :=
X SUB[ 1 , N.O.P ] LN
Y SUB[ 1 , N.O.P ] LN A.P
NEWSET SUB[ 1 , NUMBER ] LN
LOGS SUB[ 1 , NUMBER ] D.P ;
```

This program transfers the new data to a file compatible with the nonlinear regression routine. If the trailing end of the fit dips below the original data, the size of NUMBER must be reduced to eliminate this part of the gragh. It is important not to tuch the keyboard during transfer as this will jam the system.<

```
: TRANSFER
1 NEXT :=
CR ." FILENAME ?" CR
INPUT DEFER> OUT>FILE
NUMBER 1 + 1 DO
LOGS [ I ] EXP.
CR
NEWSET [ I ] .
CR
LOOP
OUT>FILE.CLOSE
;
```

APPENDIX B: NONLINEAR REGRESSION

This is a listing, in Turbo-Pascal language, of the program called EXPFIT.PAS and ran by typing expfit once it has been compiled.

The program will cease reading the file after 250 data pair of points have been entered, but this can be altered in line 4 of the program listing. The parameter estimates are fed into the computer in the following form: [1] = C1, [2] = k1, [3] = C2, [4] = K2, with the final parameter being the X-component. The final question pertaining to the restriction of eigenvalues was answered no by printing "N". An alternate criteria is incorporated but the use of its regression results is discouraged.

The NLR algorithm chosen for this work is described by Bard (70) and in the kinetic context by Mak and Langford (71). The following is a description of the NLR algorithm.

Briefly, to determine the step direction for each new iteration, the inverse scaled decomposition of the Hessian matrix N is computed. The eigenvalue decomposition of the scaled Hessian matric C was accomplished by the following sequence; (i) initial application of the Givens-House-Holder algorithm th reduce C to the tridiagonal form, (ii) diagonalization of the tridiagonal matrix by the QR algoritm with origin shifts, and (iii) successive orthogonal transformations of the unit matrix to obtain the eigenvectors.

After the step direction is established, the step length is determined by an interpolation-extrapolation algorithm. computations were terminated when a series of further iterations failed to reduce the value of the objective function which is the absolute difference between the regression model value in a previous iteration and the regression model value of the current iteration.

```
const
    matmax = 10;
    word = 16;
    maxnop = 250;
  type
   matdim = 1..matmax ;
         xary = array [ matdim,matdim ] of real ;
         yary = array [ matdim ] of real ;
         datadim = 1..maxnop ;

         zary = array [ datadim ] of real ;
  var
         i,j,size,k,m, : matdim ;
         a,v,g1,z,ninv : xary ;
         b,w,u,p,q,r,q2,v1,a1,a11,ab,ac,az : yary ;
         y,y1,aa,delta1 : zary ;
         e,e1,e2,s,x,c,d,f,h,g,c1,p1,x1,r1,d1,q1,s1,bs,
         dphi,sum,delta,deltat,z1,z2,rho,ratio : real ;
         iter : integer ;
         nop,i1 : datadim :
         choice : char ;
         test,check : boolean ;
         datain : string[ 15 ] ;
         datafile : text ;
         (*the following is a series of functions calculating the partial
         derivatives of the regression model with respect to each variable*)
  1      function derv1: real ;
  1      begin
  1      derv1:= 1-exp(-a1[2]*aa[i1])
  1      end;
  1      function derv2 ; real ;
  1      begin
  1      derv := a1{1}*aa[i1}*exp(-a1[2]*aa[i1])
  1      end ;
  1      function derv3 ; real ;
  1      begin
  1      derv3 := 1-exp(-a1[4]*aa[i1])
  1      end ;
  1      function derv4 : real ;
  1      begin
  1      derv4 := a1[3]*aa[i1]*exp(-a1[4]*aa[i1])
  1      end ;
```

```
1       function derv5 ; real ;
1       begin
1       derv5 := 1-exp(-a1[6]*aa[i1])
1       end ;
1       function derv6 : real ;
1       begin
1       derv6 := a1[5]*aa[i1]*exp(-a1[6]*aa[i1])
1       end ;
1       function derv7 : real ;
1       begin
1       derv7 := 1.0
1       end ;
1       (*the following is a series of procedures to calculate the sum of
        squares psi; and the respective gradient vector, Hessian matric etc.
        *)
1       Procedur soss (var delta : real ) ;
1       (*soss calcutates the sum of squares*)
1       begin
1               delta := 0.0 ;
1               for i1 := 1 to nop do
2               begin
3               case size of
3               3 : y1[i1] := a1[1]*(1-exp(-a1[2]*aa[i1]))+a1[3];
3               5 : y1[i1] := a1[1]*(1-exp(-a1[2]
3               *aa[i1]))+a1[3]*(1-exp(-a1[4]*aa[i1]))+a1[5];
3           7: y1[i1] := a1[1]*(1-exp(-a1[2]
3               *aa[i1]))+a1[3]*(1-exp(-a1[4]*aa[i1]))+
2               end ;
2               delta1[i1] := y[i1]-y1[i1] ;
2               delta := delta+sqr(delta1[i1]);
1               end;
        end;
1       Procedure gradient (var q2 :yary ) ;
1       (*computation of the initial gradient vector*)
1       begin
2       case size begin of
3       3 : begin
3               for i := 1 to size do
4               begin
4               q2[i] := 0.0;
4               for i1 := 1 to nop do
5               case i of
5               1 : q2[i] := q2[i]-2*delta1[i1]*derv1 ;
5               2 : q2[i] := q2[i]-2*delta1[i1]*derv2 ;
```

```
5             3 : q2[i] := q2[i]-2*delta1[i1]*derv7 ;
4             end;
3             end;
2             end;
3       5 : begin
3             for i:=1 to size do
4             begin
4             q2[i] := 0.0;
4             for i1 := 1 to nop do
5             case i of
5             1 : q2[i] := q2[i]-2*delta1[i1]*derv1;
5             2 : q2[i] := q2[i]-2*delta1[i1]*derv2;
5             3 : q2[i] := q2[i]-2*delta1[i1]*derv3;
5             4 : q2[i] := q2[i]-2*delta1[i1]*derv4;
5             5 : q2[i] := q2[i]-2*delta1[i1]*derv7
4             end;
3             end;
2             end;
3       7: begin
3             for i := 1 to size do
4             begin
4             q2[i] := 0.0;
4             for i1 :=1 to nop do
5             case i of
5             1 : q2[i] := q2[i]-2*delta1[i1]*derv1;
5             2 : q2[i] := q2[i]-2*delta1[i1]*derv2;
5             3 : q2[i] := q2[i]-2*delta1[i1]*derv3;
5             4 : q2[i] := q2[i]-2*delta1[i1]*derv4;
5             5 : q2[i] := q2[i]-2*delta1[i1]*derv5;
5             6 : q2[i] := q2[i]-2*delta1[i1]*derv6;
5             7 : q2[i] := q2[i]-2*delta1[i1]*derv7
4             end;
3             end;
2             end;
1             end;
        end;
1       Procedure Hessian (var b1 : yary; var a : xary );
1       (*computation of the initial Hessian matrix elements *)
1       begin
1             for i := 1 to size do
1             for j := 1 to size do
1             a[i,j] := 0.0 ;
1             for i1 := 1 to nop do
2             begin
```

```
3        case size of
4        3 : begin
4                 for k := 1 to size do
5                 begin
5                 z1 := 0.0;
6                 case k of
6                 1 : z1 := derv1;
6                 2 : z1 := derv2;
6                 3 : z1 := derv7;
5                 end;
5                 for m := 1 to size do
6                 begin
6                 if (m<k) then
6                 z2 := 0.0
6                 else
7                 begin
7                 z2 := 0.0
8                 case m of
8                 1 : z2 := derv1;
8                 2 : z2 := derv2;
8                 3 : z2 := derv7
7                 end;
7                 a[k,m] := a[k,m]+2*(z1*z2);
7                 a[m,k] := a[k,m];
6                 end;
5                 end;
4                 end;
3           end;
4        5 : begin
4                 for k := 1 to size do
5                 begin
5                 z1 := 0.0 ;
6                 case k of
6                 1 : z1 := derv1;
6                 2 : z1 := derv2;
6                 3 : z1 := derv3;
6                 4 : z1 := derv4;
6                 5 : z1 := derv7
5                 end;
5                 for m := 1 to size do
6                 begin
6                 if (m<k) then
6                 z2 := 0.0
6                 else
```

```
7                   begin
7                   z2 := 0.0 ;
8                   case m of
8                   1 : z2 := dervl;
8                   2 : z2 := derv2;
8                   3 : z2 := derv3;
8                   4 : z2 := derv4;
8                   5 : z2 := derv7
7                   end;
7                   a[k,m] := a[k,m]+(zl*z2);
7                   a[m,k] := a[k,m];
6                   end;
5                   end;
4                   end;
3               end;
4           7 : begin
4                   for k := 1 to size do
5                   begin
5                   zl := 0.0;
6                   case k of
6                   1 : zl := dervl;
6                   2 : zl := derv2;
6                   3 : zl := derv3;
6                   4 : zl := derv4;
6                   5 : zl := derv5;
6                   6 : zl := derv6;
6                   7 : zl := derv7
5                   end;
5                   for m := 1 to size do
6                   begin
6                   if (m<k) then
6                   z2 := 0.0
6                   else
7                   begin
7                   z2 := 0.0;
8                   case m of
9                   1 : z2 := dervl;
8                   2 : z2 := derv2;
8                   3 : z2 := derv3;
8                   4 : z2 := derv4;
8                   5 : z2 := derv5;
8                   6 : z2 := derv6;
8                   7 : z2 := derv7
7                   end;
```

```
7                     a[k,m] := a[k,m]+2*(z1*z2);
7                     a[m,k] := a[k,m];
6                     end;
5                     end;
4                     end;
3                     end;
2                     end;
1                     end;
1                     writeln;
1                     writeln;
1                     writeln (' the Hessian matrix elements row x row
                      are : ');
1                     for i := 1 to size do
2                     begin
2                     writeln;
2                     for j := 1 to size do
2                     write ( '   ',a[i,j] :9);
1                     end;
1                     (*scaling of the matrix *)
1                     for i := 1 to size do
2                     begin
2                             if (a[i,i]+0) then
2                             b1[i] := 1.0
2                             else
2                             b1[i] := sqrt(abs(a[i,i]));
1                     end;
1                     for i := 1 to size do
1                     for j := 1 to size do
2                     begin
2                             if (i=j) then
2                             a[i,j] := a[i,j]
2                             else
2                     a[i,j] := a[i,j]/sqrt(abs(a[i,i]*a[ij,j]));
1                     end;
1                     for i := 1 to size do
1                     a[i,i] := 0.0;
                      end;
1           Procedure matinv (var dphi : real ; var v1,a11 : yary)
1           (*matrix inversion begins via Givens-Householder and QR
            methods*)
1           var
1                   max,min : real ;
1           begin (*computation of e*)
1                   e := 1.0;
```

```
1              for k := 1 to word do
1              e := e*2.0;
1              e := 1.0/e;
2      begin (*computation of s, e1, and e2*)
2              s := 0.0;
2              for i := 1 to size do
2              for j := 1 to size do
2              s := s+a[i,j]*a[i,j];
2              e1 := e*sqrt(2.0*s);
2              e2 := e1/(size*size);
2              writeln;
2              writeln;
2              writeln ('the values of e1 and e2 are : ');
2              writeln (e1,'     ',e2);
1      end;
2      begin (*generate an identity matrix of SIZE*)
2              for i := 1 to size do
2              for j := 1 to size do
2              if i<>j then v[i,j] := 0.0
2              else v[i,j] := 1.0;
1      end;
2      begin
2              for i := 1 to (size-2) do
3                      begin
3                              if a[(i+1),i],i] >= 0.0 the x := 1.0

3                              else x := -1.0;
3                              c := 0.0;
3                              for j := (i+1) to size do
3                              c := c+a[j,i]*a[j,i]
3              d := c*sqrt(c);
3              b[i] := -d;
3              f := 1.0/(c+abs(a[(i+1),i]*d));
3              w[i+1] := a[(i+1),i]+d;
3              for j := (i+2) to size do
3              w[j] := a[j,i];
3              for j := (i+1) to size do
3              u[j] := f*w[j];
3              for k := 1 to size do
4                      begin
4                      p[k] := 0.0;
4                      for j := (i+j) to size do
4                      p[k] := p[k]+v[k,j]*w[j];
3                      end;
```

```
3     for k := i to size do
3     for j := (i+1) to size do
3     v[k,j] := v[k,j]-p[k]*u[j];
3     for k := (i+1) to size do
4             begin
4             q[k] := 0.0;
4             for j := (i+1) to size do
4             q[k] := q[k]+a[k,j]*u[j];
3             end;
3     h := 0.0;
3     for k := (i+1) to size do
3     h := h+q[k]*u[k];
3     h := h/2.0;
3     for k := (i+1) to size do
3     q[k] := q[k]-h*w[k];
3     for j := (i+1) to size do
3     for k := (i+1) to size do
3     a[j,k] := a[j,k]-q[j]*w[k]-w[j]*q[k];
2     end;
2     writeln;
2     writeln;
2     writeln ('the numerical elements of the final tridiagonal
      matrix are ; ');
2     writeln;
3     for i := 1 to size do
3     begin
3     writeln;
3     for j := 1 to size do
3     write ('  ',a[i,j] ; 9);
2     end;
2     (*QR decomposition*)
2     b[size-1] := a[(size-1),size];
2     for i := 1 to size do
2     r[i] := a[i,i];
2     m := size
2     g := 0.0 ;
3     repeat
3     test := true;
3     c1 := 1.0;
3     f := r[m];
3     p1 := r[1];
3     for i := 1 to (m-1) do
4             begin
4             if abs(b[i]) <= e2
```

```
5            then begin
5            b[i] := 0.0;
5            d := 0.0;
5            c := p1;
4            end;
5            else begin
5                x1 := sqrt(p1*p1+b[i]*b[i]);
5                d := b[i]/x1;
5                c := p1/x1;
5                for j :=1 to size do
6                    begin
6                    h := c*v[j,(i+1)]-d*v[j,i];
6                    v[j,i] := d*v[j,(i+1)]+c*v[j,i];
6                    v[j,(i+1)] := h;
5                    end;
5                end;
4            r1 := c*p1+d*b[i];
4            d1 := c *c1;
4            q1 := d1*b[i]+d*r[i+1];
4            r[i] := d1*r1+d*q1;
4            if i > 1 then
4            b[i-1] := s1*r1;
4            s1 :=d;
4            p1 := c*r[i+1}-s1*c1*b[i];
4            c1 := c;
3            end;
3        b[m-1] := s1*p1;
3        r[m] := c1*p1;
3        bs := 0.0;
3        for k :=1 to (m-1) do
3        bs := bs+abs(b[k]);
3        if (bs <= e1) and (test = true)
4        then begin
4            for k :=1 to m do
4            r[k] := r[k]+g;
4            test := false;
3            end;
3        while (abs(b[m-1]) <= e2) and (test=true) do
4            begin
4            if m >= 2 then
5            begin

5            r[m] := r[m]+g;
5            m := m-1;
```

```
4        end
4        else
5            begin
5            if (abs(abs(r[m]/f)-1) > 0.5) and (test=true) then
5            test := true
5            else
6            begin
6            g := g+r[m];
6            for k := 1 to m do
6            r[k] := r[k]-r[m];
5            end
4            end;
4            if m >= 2 then
4            test := true
4            else
4            test := false;
3            end;
2            until test := false;
2            (*Eigenvalue adjustments if necessary begins*)
2            for i := 1 to size do
2            r[i] := abs(r[i]);
2            if choice = 'y' then
3            begin
3            max := r[i];
3            min := r[i];
3            for i := 2 to size do
4            begin
4            if r[i]>max then
4            max := r[i];
4            if r[i]<min then
4            min := r[i];
3            end;
3            if (min/max)<0.000005 then
4            begin
4            writeln;
4            writeln;
4            writeln ('the Hessian matrix or the initial estimates are ill-conditioned');
4            writeln;
4            writeln ('the value of max/min is :', max/min);
4            for i := 1 to size do
```

```
4                 r[i] := 0.000005*max+r[i];
3                 end;
2                 end;
2                 writeln;
2                 writeln;
2                 writeln ('the eigenvalue are:');
2                 for k := 1 to size do
2                 write (' ',r[k] :9);
2                 writeln;
2                 writeln;
2                 writeln ('the eigenvectors are:');
2                 for i :+ 1 to size do
3                 begin
3                 writeln;
3                 for j := 1 to size do
3                 write (' ',v[j,i] : 9) :
2                 end
1         end;
1         for i := 1 to size do
1         for j := 1 to size do
1         g1[i,j] := 1/b1[i]*v[i,j];
1         for i := 1 to size do
1         for j := 1 to size do
1         z[i,j] := g1[i,j]*1/r[j];
1         for i := 1 to size do
2         begin
2                 sum := 0.0;
2                 for k := 1 to size do
3                 begin
3                 sum := 0.0;
3                 for j := 1 to size do
3                 sum := sum+z[i,j]*g1[k,j];
3                 ninv[i,k] := sum;
2                 end;
1         end;
1         writeln;
1         writeln;
1         writeln('the values of ninv[i,k] are :');
1         for i := 1 to size do
2         begin
2         writeln;
2         for j := 1 to size do
2         write ('   ',ninv[i,j] : 9);
1         end;
```

```
1               for i := 1 to size do
2               begin
2                       v1[i] :=0.0;
2                       for j := 1 to size do
2                       v1[i] :- v1[i]+ninv[i,j]*(-q2[j]);
1               end;
1               writeln;
1               writeln;
1               writeln ('the values of v1[i] are : ')'
1               for k := 1 to size do
1               writeln ('v1[',k,'] = ',v1[k] : 9);
1               dphi := 0.0;
1               for i := 1 to size do
1               dphi := dphi+v1[i]*q2[i];
1               writeln;
1               writeln;
1               writeln ('the value of "dphi" is : ');
1               write (dphi :9);
1               for i :=1 to size do
1               a11[i] := a1[i];
1               end;
1       procedure stepsize (var a1: yary; var check : boolean );
1       label 99
1       var
1       ab: yary ;
1       maxo,max dummy,max1,delta1,min : real ;
1       count,index : integer;
1       size1 : matdim;
1       (*the following function NRHO is to compute
1        the stepsize proportionality factor*)
2       function nrho : real;
2       begin
2       nrho := dphi*sqr(rho)/(2*(dphi*rho+deltat-delta))
1       end;
2       procedure doss (var delta ; real);
2       var
2               i1 : datadim;
2       begin
2               delta := 0.0;
2               for i1 := 1 to nop do
3                       begin
4                       case size of
4                       3   :   y1[i1]   :=   a1[1]*(1-exp(-a1[2]
                        *aa[i1]))+a1[3];
```

```
4                       5 : yl[il] := al[i]*(1-exp(-al[2]*
4                       aa[il]))+al[3]*(1-exp(-al[4]* aa[il]))+al[5];
4                       7 : yl[il] := al[1]*(1-exp(-al[1]*
4                       aa[il]))+al[5]*(1-exp(-al[6]* aa[il]))+al[7];
3                       end;
3                       deltal[il] := y[il]-yl[il];
3                       delta := delta+sqr(deltal[il]);
2                       end;
1       end;
1       (*the main procedure stepsiz begins below *)
1       begin
1               check := true;
1               count := 0 ;
1               max1 := 1.0;

2       repeat
2       count := count+2;
2       if (abs(v1[count])>max1 then
2       max1 := abs(v1[count]);
1       until count := size-1;
1       dummy := 1.0;
1       maxo := 0.5;
1       if max1>0.5 then
2       repeat
2       dummy := dummy*0.5;
2       max0 := max1*dummy;
1       until maxo<=0.5;
1       writeln;
1       writeln;
1       writeln;
1       writeln ('the initial adjusted a1[i] values are : ");
1       writeln;
1       for i := 1 to size do
2       begin
2       v1[i] := v1[i]*dummy;
2       a1[i] := a1[i]+v1[i];
2       writeln ('a1[i,'i,'] = ',a1[i] : 9);
1       end;
1       deltat := delta;
1       doss(delta ;
1       min := 0.0 ;
1       for i := 1 to size do
1       if a1[i]<min then
1       min := a1[i];
```

```
1            if (delta<deltat) and (min>0.0) then
2            begin
2            writeln;
2            writeln ('the initially adjusted a1[i] values are used
             directly');
2       writeln (for extrapolation/interpolation');
1            end
1            (*Greenstadt Type adjustments begins*)
1            else
2            begin
2            for i := 1 to size do
2            ab[i] := a1[i];
2            delta1 := deltat;
2            index := 0.0;
2            count := 0 ;
3            repeat
3            count := count+2;
3            if ab[count]<0.0 then
4            begin
4            a1[count] := abs(ab[count]*0.15);
4            index := index+1;
3            end;
3            if ab[count-1]<0.0 then
4            begin
4            a1[count-1] := abs(ab[count-1]*0.75);
4            index := index+1;
3            end;
2            until count = size-1;
2            if ab[size]<0.0 then
3            begin
3            a1[size] := abs(ab[size]*0.5);
3            index := index+1;
2            end;
2            for i := 1 to size do
3            begin
3            if ab[i]>0.0 then
3            a1[i] := a1[i]*(index*0.15);
2            end;
2            doss(delta);
2            if delta<delta1 then
3            begin
3            writeln;
3            writeln ('the "scaled" values of a1[i] are : ') ;
3            for i := 1 to size do
```

```
3    writeln ('scaled a1[i] =',a1[i] :9);
2    end
2    else
3    begin
3    count := 0.0;
4    repeat
4    delta1 := delta;
4    count := count+1;
5    case size of
5    3 : size1 := 1;
5    5 : size1 := 2;
5    7 : size! := 3;
4    end;
4    for i := 1 to size1 do
5    begin
5    if ab[i*2]<0.0 then
5    a1[i*2] := a1[i*2]+abs(ab[i*2]*0.1);
4    end;
4    doss(delta);
3    until (delta<delta1) or (count=8);
3    if delta1<=delta then
4    begin
5    case size of
5    3 : size1 := 2;
5    5 : size1 := 3;
5    7 : size1 := 4
4    end;
4    for i := 1 to size1 do
5    begin
5    if ab[2*i-1]<0.0 then
5    a1[2*i-1] := a1[2*i-1]*1.2;
4    end;
3    end;
3    delta1 := delta;
3    doss(delta);
3    if delta1<delta then
4    begin
4    for i := 1 to size1 do
5    begin
5    if ab[2*i-1]<0.0 then
5    a1[2*i-1] := a1[2*i-1]*(2/3);
4    end;
3    end;
3    delta1 := delta;
```

```
3    doss(delta);
3    if delta<delta then
4    begin
5    case size of
5    3 : size1 := 1;
5    5 : size1 := 2;
5    7 : size1 := 3
4    end;
4    for i := 1 to size1 do
5    begin
5    if ab[2*i-1]<0.0 then
5    a1[2*i-1] := abs(ab[2*i-1]*0.5);
5    if ab[2*i]<0.0 then
5    a1[2*i] := abs(ab[2*i]*0.15);
4    end;
4    if ab[size]<0.0 then
4    a1[size] := abs(ab[size]*0.5);
3    end;
3    writeln;
3    writeln ('the "scaled" values of a1[i] are : ');
3    for i := 1 to size do
3    writeln ('scaled a1[i] = ', a1[i] : 9 );
2    end;
1    end;
1    for i := 1 to size do
1    az[i] := 0.0001*(a1[i]+0.001);
1    ratio := az[i]/abs(v1[1]);
1    for i := 2 to size do
2    begin
2    if (az[i]/abs(v1[i]))<ratio then
2    ratio := az[i]/abs(v1[i]);
1    end;
1    if abs(deltat-delta)<1.0e-08 then
2    begin
2    check := false;
2    goto 99;
1    end
1    else
2    begin
2    writeln;
2    writeln ('the boolean value of delta and deltat are:');
1    end;
1    rho := 1.0;
1    writeln;
```

```
1          writeln ('the new values of delta and deltat are :');
1          writeln (delta : 9,'     ',deltat : 9);
1          if delta<deltat then
2          begin
2          if nrho<0 then
2          dummy := 2*rho
2          else
2          dummy := nrho;
2          writeln;
2          writeln ('the value of dummy is :') n;
2          writeln (dummy :9);
2          if abs(dummy-rho)<(0.1*rho) then
2          dummy := rho
2          else
2          dummy := dummy;
2          for i := 1 to size do
2          a1[i] := abs(a11[i]+dummy*v1[i]);
2          doss(delta);
2          if delta>deltat then
2          dummy := rho
2          else
2          dummy := dummy;
2          for i :=1 to size do
3          begin
3          a1[i] := abs(a11[i]+dummy*v1[i]);
3          a11[i] := a1[i];
2          end;
2          writeln;
2          writeln ('the a1[i] values by extrapolation
           are : ');
2          for i := 1 to size do
2          writeln ('EXT a1[',i,'] = '.a1[i] :9);
1          end
1          else
1          writeln;
1          writeln;
1          writeln ('interpolation begins at this point  ;');
2          begin
2          count := 0.0;
3          repeat
3          count := count+1;
3          writeln;
3          writeln ('the value of rho and nrho respectively are :');
3          writeln ('rho= ',rho, '     ','nrho =',nrho);
```

```
3    if nrho<(0.75*rho) then
3    dummy := nrho
3    else
3    dummy := 0.75*rho;
3    if dummy<(0.25*rho) then
3    dummy := 0.25*rho
3    else
3    dummy := dummy;
3    for i := 1 to size do
3    a1[i] := abs(a11[i]+dummy*v1[i]);
3    doss(delta);
3    rho := dummy;
3    writeln;
3    writeln ('the current interpolation cycle is : ', count);
3    writeln ('the values of a1[i] by interpolation are :');
3    for i := 1 to size do
3    writeln ('INT a1['.i,'] = ',a1[i] :9);
2    until (delta<deltat) or (count =15) ;
2    if ratio>abs(nrho) then
3    begin
3    check := false;
3    writeln;
3    writeln;
3    writeln ('iterations terminated by way fo alternative criterion');
2    end;
1    end;
     99 : end;
     begin
     (*main programme begins*)
     writeln;
     writeln ('what is the name of your data file ?');
     read (datain);
     writeln;
     writeln;
     writeln ('input the number of parameters to be fitted or SIZE :');
     read (size);
     writeln;
     writeln;
     writeln ('input initial estimates of parameters; i.e.,a[1]...a[size]');
     for i := 1 to size do
     begin
```

```
1      writeln ('Enter a1[',i,'] : ');
1      readln (a1[i]);

       writeln;
       writeln ('do you wish to restrict eigenvalues? - answer "y" or "n"');
       read(choice);
       writeln;
       writeln;
       writeln ('the values of the initial estimates are :');
       for i := 1 to size do
1      begin
1      ac[i] := a1[i];
1      writeln ('a1[',i,'] = ',ac[i] :9);
       end;
       assign (datafile,datain);
1      begin
1      reset (datafile);
1      nop := 0.0;
1      while not eof(datafile) and (nop<maxnop) do
2      begin
2      while not eoln(datafile) and (nop<maxnop) do
3      begin
3      nop := nop+1;
3      read(datafile,y[n.o.p],aa[nop]
2      end;
2      readln(datafile);
1      end;
1      close (datafile);
       end;
       writeln;
       writeln;
       writeln ('the value of nop is :');
       writeln (nop);
       writeln;
       writeln;
       writeln ('the values of y[i1]s, and aa[i] are : ' );
       for i := 1 to nop do
       writeln (y[i],'   ',aa[i]);
       iter := 0.0;
       repeat
1      iter := iter+1;
1      check := true;
1      soss(delta;
1          writeln;
1          writeln;
1          writeln ('the value of delta/sum of squares is : ');
1          gradient(q2);
1          writeln;
1          writeln;
1          writeln ('the gradient vector of dimension size is :');
1          for i := 1 to size do
1          writeln ('q2[',i,'] = ',q2[i] ; 9);
1          hessian(b1,a);
           writeln;
           writeln;
           writeln ('the diagonal elemints of matrix B1 is : ');
1          for i := 1 to size do
1          writeln ('b1[',i'] = ',b1[i] :9);
           writeln;
1          writeln;
1          writeln ('the elements of matrix A is : ');
           for i := 1 to size do
           begin
           writeln;
2          for j := 1 to size do
2          write ('   '.a[i,j] : 9);
1          end;
1          matinv (dphi,v1,a11);
           stepsiz (a1,check);
           until check := false;
           writeln;
           writeln;
           writeln ('iterations terminated/boolean check is : ', check );
           writeln ('the final values of the regression parameters are :
           ');
           for i := 1 to size do
           writeln ('a1[',i,'],a1[i] :9);
           writeln;
           writeln;
           writeln ('the original parameters estimates were : ' ):
           for i := 1 to size do
           writeln ('a1[',i,'] =',a1[i] :9);
           writeln;
           writeln ('number of iterations thus far :')
           writeln (iter)
           end.
```