

Host/Guest Interactions and NMR Spectroscopy. A Computer Program for Association Constant Determination

DOLORS SALVATIERRA, CARLES DíEZ and CARLOS JAIME*

Departament de Química, Facultat de Ciències, Universitat Autònoma de Barcelona, 08193 Bellaterra, Spain.

(Received: 2 April 1996; in final form: 17 July 1996)

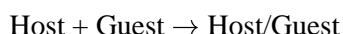
Abstract. A computer program is presented for the computation of association constants for host/guest compounds based on experimental data obtained from the NMR spectra of samples with different host/guest ratios of known concentration.

Key words: Association constant, inclusion complexes, host/guest chemistry.

1. Introduction

Research in host/guest chemistry should follow several well defined steps: proof of the inclusion, determination of the complex stoichiometry, determination of the association constant, and, eventually, determination of the inclusion complex geometry. While the first two steps are appropriately solved by experiments, the other two steps are not so well treated in the scientific literature.

Many experimental techniques allow confirmation of the formation of an inclusion complex, and descriptions exist in the literature of procedures for determining the complex stoichiometry. Among these, the Job [1] diagram is probably the most reliable. The determination of the association constant has been widely treated and there are several methods for its determination in a 1 : 1 complex; all based on the consideration of the general equation:



Developing the association constant (K) expression and considering p as the molar fraction of the complexed guest, K adopts the following expression:

$$K = \frac{p}{([\text{host}]_i - p[\text{guest}]_i)(1 - p)}$$

NMR spectroscopy is nowadays probably one of the most widely used techniques for host/guest studies. The NMR parameters (δ and J) are usually weighted averages

* Author for correspondence.

from all species present; in consequence, when the host/guest complex is in fast equilibrium with the isolated molecules, the experimental δ_{obs} corresponding to an intermediate situation can be expressed by the weighted average between the isolated (δ_f) and the complexed (δ_c) chemical shifts. Moreover, the molar fraction of any component can be expressed as the ratio between $\Delta\delta_i$ ($\delta_{\text{obs}} - \delta_f$) and $\Delta\delta_c$ ($\delta_c - \delta_f$). Introducing these two assumptions into the expression for K , it can be transformed into:

$$K = \frac{\Delta\delta_i/\Delta\delta_c}{([\text{host}]_i - (\Delta\delta_i/\Delta\delta_c)[\text{guest}]_i)(1 - (\Delta\delta_i/\Delta\delta_c))} \quad (1)$$

The experimental determination of association constants in host/guest compounds usually contains some approximations. The methods due to Scatchard [2], Benesi and Hildebrand [3], and Scott [4] are the most commonly used for the determination of K . All require the graphical representation of experimental parameters and the global consideration of $[\text{host}]_i \gg [\text{guest}]_i$. The graphical representation of data is considered to be an advantage, while the use of data where the concentration for the host has to be much larger than that for the guest is not always either possible or mathematically correct. Experimentally, $\Delta\delta_i$ is determined for each of the studied protons and, consequently, the use of the corresponding expression for K yields a different K value for each one of the studied protons.

In this article we would like to present the development of a simple computer program to be used with the whole set of studied protons, producing one single K value for the whole process and a set of calculated $\Delta\delta_c$ values.

2. Results and Discussion

The systematic variation of the chemical shifts for some NMR signals is the data usually employed as unequivocal proof for the host/guest interaction. The methods used for the determination of K need to fulfil the requirement of $[\text{host}]_i \gg [\text{guest}]_i$. When this condition is satisfied, the $[\text{guest}]_i$ is usually very small and the difficulty in determining the chemical shift for a signal increases with its complexity (coupling). In addition, the standard range of work only permits host/guest ratios not usually larger than 9/1, a ratio which mathematically cannot be considered as satisfying $[\text{host}]_i \gg [\text{guest}]_i$.

The development of Equation (1), using A as the $[\text{host}]$ and B as the $[\text{guest}]$, allows us to obtain a second order equation whose solution is Equation (2).

$$\Delta\delta_i = \frac{\Delta\delta_c}{2B_i} \left[\left(A_i + B_i + \frac{1}{K} \right) \pm \sqrt{\left(A_i + B_i + \frac{1}{K} \right)^2 - 4A_iB_i} \right]. \quad (2)$$

Only the negative square root solution is to be considered because the ratio $\Delta\delta_i/\Delta\delta_c$ is always smaller than 1. Moreover, the definition of p restricts the number of

samples to be used because now $[\text{host}]_i$ should be only greater than $[\text{guest}]_i$ (i.e., $A_i > B_i$) instead of being much greater, and this fact can be accomplished in about one half of the samples studied.

If we define a new variable, l , as:

$$l_i = \left(A_i + B_i + \frac{1}{K} \right) - \sqrt{\left(A_i + B_i + \frac{1}{K} \right)^2 - 4A_i B_i}, \quad (3)$$

then Equation (2) is transformed into a simpler one:

$$\Delta\delta_i = \frac{\Delta\delta_c}{2B_i} l_i. \quad (4)$$

This equation presents two variables ($\Delta\delta_c$) and l_i (which is a function of K). Equation (4) must be satisfied for each sample studied, i.e., we have a set of n equations (as many as samples) with two independent variables ($\Delta\delta_c$ and K). The equation can be solved by using the least squares approximation using the experimental $\Delta\delta_i$ values as reference values. Obviously, there is not one single pair of $\Delta\delta_c/K$ values which will verify Equation (4) for all the existing experimental $\Delta\delta_i$ values corresponding to a determined studied proton.

The best solution was defined as the pair making minimum the first derivative of the least squares function with respect to each independent variable. This mathematical treatment allows us to rewrite Equations (5) and (6).

$$\frac{\partial E}{\partial \Delta\delta_c} = 0 \Rightarrow \Delta\delta_c \sum \frac{l_i^2}{4B_i^2} = \sum \frac{\Delta\delta_i l_i}{2B_i} \quad (5)$$

$$\frac{\partial E}{\partial K_c} = 0 \Rightarrow \Delta\delta_c \sum \frac{l_i^2}{2B_i \left(A_i + B_i + \frac{1}{K} - l_i \right)} = \sum \frac{\Delta\delta_i l_i}{A_i + B_i + \frac{1}{K} - l_i} \quad (6)$$

By isolation of $\Delta\delta_c$ from Equation (5) and its substitution in Equation (6) a new equation is obtained with only one independent variable, K , which can be solved by the Newton–Raphson method [5]. Strictly speaking, we should obtain one equation (or one K) for each of the studied protons of the guest (each set of $\Delta\delta_i$).

Following an analogous procedure but now defining p as the molar fraction of complexed host, equations equivalent to 5 and 6, but now for the host protons, are obtained. The process for the formation of an inclusion complex should be unique for all the protons of the system studied and, in consequence, only one single K with a set of $\Delta\delta_c$ values (one for each host and guest protons studied) should be obtained. The treatment of the whole set of protons studied yields Equation (7), which represents the least squares deviation for the whole process. The Newton–Raphson method applied to this new equation will afford the expected single K value and the set of $\Delta\delta_c$ values.

$$E = \sum_{j=1}^m \left[\sum_{i=1}^n \left(\Delta\delta_i - \frac{\Delta\delta_c}{2A_i} l_i \right)^2 \right]. \quad (7)$$

Table I. Host and guest concentrations (mol/L) and induced chemical shifts ($\Delta\delta$) for each of the samples used in the study of the benzoic acid/ β -cyclodextrin complex [6].

Sample	[host] $\times 10^2$	[guest] $\times 10^2$	$\Delta\delta_i$				
			H- <i>o</i>	H- <i>m</i>	H- <i>p</i>	H-3	H-5
1	1.3040	0.3771	-0.0429	0.0188	-	0.0000	0.0000
2	1.0595	0.5679	-0.0342	0.0100	-0.0086	-0.0263	-0.0332
3	0.8965	0.7828	-0.0240	0.0066	-0.0072	-0.0484	-0.0635
4	0.7335	0.8608	-0.0191	0.0051	-0.0057	-0.0616	-0.0812
5	0.5705	1.0950	-0.0079	0.0046	-0.0018	-0.0748	-0.1027
6	0.2445	1.1660	0.0000	0.0000	0.0000	-0.0829	-0.1230

To solve Equation (7) a computer program (called CALCK) has been developed, written in Fortran (see Appendix), which as input will use the number of protons studied, the number of samples for each studied proton and $\Delta\delta_i$, $[\text{guest}]_i$, and $[\text{host}]_i$. The value for K and the $\Delta\delta_c$ for each studied proton will form the output.

It should be noted here that the general behaviour of functions like the first derivative of Equation (7) have a zone where the function presents numerous artificial solutions, when K is very small. After this zone, the function takes negative values and starts to grow slowly until it crosses the axis, giving a single zero value which is the real solution of Equation (7). Thereafter, the function increases and diverges.

This behaviour forces the use of several test entries for K contained in an auxiliary file called CONST.DAT. The program changes the K values automatically and will find the real solution for the variable.

3. Illustrative Example

The complex between benzoic acid and β -cyclodextrin has been studied [6]. Table I contains host and guest concentrations, and induced chemical shifts (referred to the position of the cyclodextrin anomeric proton) observed for each of the studied protons (H-*o*, H-*m*, and H-*p* for benzoic acid and H-3 and H-5 for cyclodextrin) in each sample. The input file (INPUT.DAT) for this case is shown in Figure 1 and part of the output file (OUTPUT.OUT) is contained in Figure 2. The calculated association constant for this complex is 48.67 l/mol.

Acknowledgements

Professor J. Elguero, from the Institute of Medicinal Chemistry, CSIC, Madrid, is thanked for helpful discussions and suggestions. The DGICYT (project no. PB92-0611), Ministerio de Educación y Ciencia, Madrid, Spain, is gratefully acknowledged for financial support.

'BENZOIC AC.'	26
5	1d-2
'H-3B'	1d-1
3	1.d0
0.7335d-2 0.8608d-2 0.0616	1.d1
0.5705d-2 1.095d-2 0.0748	1.d2
0.2445d-2 1.166d-2 0.0829	2.d2
1.D-6 1.7d4	3.d2
1.D-15	4.d2
100	5.d2
'H-5B'	6.d2
3	7.d2
0.7335d-2 0.8608d-2 0.0812	8.d2
0.5705d-2 1.095d-2 0.1027	9.d2
0.2445d-2 1.166d-2 0.1230	1000d0
1.D-6 1.7d4	2000d0
1.D-15	3000d0
100	4000d0
'H-2 '	5000d0
3	6000d0
0.3771d-2 1.3040d-2 0.0429	7000d0
0.5679d-2 1.0595d-2 0.0342	8000d0
0.7828d-2 0.8965d-2 0.0240	9000d0
1.D-6 1.7d4	1.d4
1.D-15	5.d4
100	1.d5
'H-3'	5.d5
3	
0.3771d-2 1.304d-2 0.0188	*****
0.5679d-2 1.0595d-2 0.0100	
0.7828d-2 0.8965d-2 0.0066	
1.D-6 1.7d4	
1.D-15	
100	
'H-4'	
2	
0.5679d-2 1.0595d-2 0.0086	
0.7828d-2 0.8965d-2 0.0072	
1.D-6 1.7d4	
1.D-15	
100	
0.070d0 1.d0	

(A)**(B)**

Figure 1. Example of INPUT.DAT (A) and CONST.DAT (B) used in the study of the benzoic acid/ β -cyclodextrin complex.

COMPLEX: BENZOIC AC

** VALUES OF R. EXPLANATION:

- 1 - OK (MINIMUM)
- 2 - NOT OK. (MAXIMUM)
- 3 - NOT OK. (MINIMUM BUT DELTAC
SMALLER THAN EXPERIMENT)
- 9 - NOT OK. (NEGATIVES SOLUTIONS)

COMPLEX: BENZOIC AC

- 1= PROTON:H-3B
- 2= PROTON:H-5B
- 3= PROTON:H-2
- 4= PROTON:H-3
- 5= PROTON:H-4

J K INI. ASSOC. CONSTANT CALCD DELTA ERROR FUNC R

***** SEVERAL LINES HAVE BEEN REMOVED *****

```

1 0.1E+02 -0.2199407787E-09 -0.56272E-13 0.52963E-02 3
2 0.1E+02 -0.2199407787E-09 -0.80606E-13 0.52963E-02 3
3 0.1E+02 -0.2199407787E-09 -0.38258E-13 0.52963E-02 3
4 0.1E+02 -0.2199407787E-09 -0.14188E-13 0.52963E-02 3
5 0.1E+02 -0.2199407787E-09 -0.11312E-13 0.52963E-02 3

```

```

1 0.1E+03 0.4867076445E+02 0.24268E+00 0.11081E-03 1
2 0.1E+03 0.4867076445E+02 0.34163E+00 0.11081E-03 1
3 0.1E+03 0.4867076445E+02 0.11146E+00 0.11081E-03 1
4 0.1E+03 0.4867076445E+02 0.40129E-01 0.11081E-03 1
5 0.1E+03 0.4867076445E+02 0.28454E-01 0.11081E-03 1

```

```

1 0.2E+03 0.4867076445E+02 0.24268E+00 0.11081E-03 1
2 0.2E+03 0.4867076445E+02 0.34163E+00 0.11081E-03 1
3 0.2E+03 0.4867076445E+02 0.11146E+00 0.11081E-03 1
4 0.2E+03 0.4867076445E+02 0.40129E-01 0.11081E-03 1
5 0.2E+03 0.4867076445E+02 0.28454E-01 0.11081E-03 1

```

```

1 0.3E+03 0.4867076445E+02 0.24268E+00 0.11081E-03 1
2 0.3E+03 0.4867076445E+02 0.34163E+00 0.11081E-03 1
3 0.3E+03 0.4867076445E+02 0.11146E+00 0.11081E-03 1
4 0.3E+03 0.4867076445E+02 0.40129E-01 0.11081E-03 1
5 0.3E+03 0.4867076445E+02 0.28454E-01 0.11081E-03 1

```

***** SEVERAL LINES FOLLOW BUT ARE NOT INCLUDED *****

Figure 2. Example of output file (only parts of OUTPUT.OUT) obtained in the study of the benzoic acid/ β -cyclodextrin complex.

Appendix

```

C
C
C   PROGRAM CALCK
C
C   PROGRAM FOR OBTAINING THE ASSOCIATION CONSTANT BY
C   NEWTON-RAPHSON (N-R) METHOD WITH COMPUTATION OF DELTAC
C
C   WORKS WITH ALL PROTONS AT SAME TIME (UP TO 15)
C
C
C
C   PROGRAM GOAL:
C   =====
C
C   THIS PROGRAM TRIES TO FIND ASSOCIATION CONSTANTS FOR
C   COMPLEXES USING SETS OF VALUES FOR THE INITIAL CONCENTRATIONS
C   OF PRODUCTS AND VARIATIONS OF CHEMICAL SHIFTS.
C
C   PROGRAM ROUTINES:
C   =====
C
C   MAIN: READS THE VARIABLES AND DOES THE LOOP FOR EACH
C   INITIAL K. EACH ITERATION OF THIS LOOP IS AN EXECUTION OF N-R.
C
C   CALCULF: FINDS THE VALUES OF FUNCTIONS F AND DF WHICH
C   DUE TO THEIR COMPLEXITY WILL BE SOLVED IN ANOTHER
C   ROUTINE USING AUXILIARY VARIABLES.
C
C   ORIGINAL: FINDS THE VALUE FOR FUNCTION FORIG
C
C   ESCRIURE: WORKING ROUTINE FOR THE PRINTING OF VARIABLES. THIS ROUTINE
C   HAS BEEN MAINLY USED FOR THE PROGRAM DEBUGING.
C   ON THE LAST STATE OF THE PROGRAM IT DOESN'T HAVE ROUTINE ENTITY.
C
C   RELLEVANT VARIABLES:
C   =====
C
C   INTERNAL VARIABLES USED IN EACH ROUTINE ARE NOT CONSIDERED.
C   NEITHER WORKING VARIABLES FOR THE PROGRAM.
C
C   NKS:           NUMBER OF INITIAL K VALUES TO BE USED IN THE ITERATIONS
C   KINI(NKS):     INITIAL K VALUES
C   NMOL:          IDENTIFICATION NAME (FOR THE PRINTING)
C   NPROT:         PROTON NAME (FOR THE PRINTING)
C   NOBS:          NUMBER OF SAMPLES
C   A(NOBS):       CONCENTRATION OF A
C   B(NOBS):       CONCENTRATION OF B
C   DELTA(NOBS):  VARIATIONS OF CHEMICAL SHIFT DETECTED
C   TOL:           FIXED TOLERANCE TRIED FOR THIS EXECUTION
C   NMAX:          MAXIMUM NUMBER OF ITERATIONS ALLOWED FOR EACH
C                 EXECUTION
C   FORIG:         ERROR FUNCTION (FOR MINIMIZATION)
C   F:             PARTIAL DERIVATIVE OF FORIG RELATED TO K.
C   DF:           PARTIAL DERIVATIVE OF F RELATED TO K
C
C
C

```

```

C FILES DESCRIPTION:
C =====
C
C INPUT:
C
C 4 CONST.DAT: INITIAL SET OF TRIAL VALUES OF K FOR RUNNING N-R METHOD
C   LINE 1:      NKS   : NUMBER OF INITIAL K TO TRY
C   LINE 2 A NKS+1: KINI(I) : INITIAL K VALUES
C
C 5 INPUT.DAT: PROGRAM DATA:
C   LINE 1:  NMOL   : COMPLEX NAME (FOR RESULTS PRINTING)
C   LINE 2:  NPROT  : PROTON NAME (FOR RESULTS PRINTING)
C   LINE 3:  NOBS   : NUMBER OF SAMPLES
C   LINES 4 A NOBS+3:  A(I)   : CONCENTRATION OF A
C                   B(I)   : CONCENTRATION OF B
C                   DELTA(I): CHEMICAL SHIFTS VARIATIONS
C   LINE NOBS+4:  KMIN  : THIS VALUE IS NOT USED WITHIN THIS PROGRAM:
C                   IT REMAINS FOR COMPATIBILITY WITH THE
C                   SOFTWARE.
C                   KMAX   : IDEM
C   LINE NOBS+5:  TOL   : TOLERANCE FIXED FOR THIS EXECUTION
C   LINE NOBS+6:  NMAX  : MAXIMUM ITERATIONS ALLOWED FOR EACH N-R
C                   EXECUTION
C
C OUTPUT:
C
C 11 NEWTON.RES:  FILE WHERE PROGRAM WRITES THE RESULT OF
C                 EACH ITERATION OF N-R.
C
C 12 OUTPUT.OUT: TABLE A WHERE HAVING A RUNNING SUMMARY WRITTING:
C                 INITIAL K USED
C                 ASSOSIATION CONSTANT FOUND
C                 DELTAC FOUND
C                 VALUE FOR THE FUNTION ERROR FORIG
C                 RUNNING RESULT, IT CAN BE:
C                 1 - OK (MINIMUM)
C                 2 - NOK. (MAXIMUM)
C                 3 - NOK. (MINIMUM BUT DELTAC SMALLER THAN MAXIMUM)
C                 9 - NOK. (THERE ARE NEGATIVE SOLUTIONS).
C
C 6: DISPLAY:     INFORMATION FOR THE USER ABOUT, FOR EXAMPLE,
C                 RESULTS LIKE TIPE "9". THIS INFORMATION IS ALSO ON FILE 11.
C
C WORKING WITH UP TO 15 PROTONS
C
C DECLARATION OF VARIABLES
C
REAL*8  TOL,A(15,100),B(15,100),DELTA(15,100),INCREM
REAL*8  KMIN,KMAX,FORIG,DELTAC(15)
REAL*8  KINI(100)
REAL*8  KI,F,DF
REAL*8  DELTAMIN(15),DELTAMAX(15)
REAL*8  P,KTEOR
INTEGER*4 I,J,NOBS(15),NMAX,COUNT,NKS,IRES,IERR,NUPRO
CHARACTER NOMPROT*10,NOMMOL*10
C
C FILE'S DECLARATIONS

```



```

C
OPEN(4,FILE='CONST.DAT')
OPEN(5,FILE='INPUT.DAT')
OPEN(11,FILE='NEWTON.RES')
OPEN(12,FILE='OUTPUT.OUT')
C
C INITIALIZATIONS
C
DO 77 J=1,15
  NOBS(J)=0
  DELTAC(J)=0.D0
  DO 7 I=1,100
    A(J,I)=0.D0
    B(J,I)=0.D0
    DELTA(J,I)=0.D0
  7 CONTINUE
77 CONTINUE
C
C READING VARIABLES. FROM FILE 4 (CONST.DAT) READ INITIAL K VALUES
C
WRITE(11,*) 'INPUT NUMBER OF K"S'
READ(4,*,ERR=101) NKS
WRITE(11,*) 'NO. OF INITIAL K:',NKS
IF (NKS .GT. 100) THEN
  WRITE(11,*) 'NUMBER OF KS INIT. > 100: RECOMPILE THE PROGRAM'
  GOTO 101
ENDIF
WRITE(11,*) 'INPUT INITIAL VALUES FOR K'
DO 10 I=1,NKS
  READ(4,*,ERR=101) KINI(I)
  WRITE(11,*) KINI(I)
10 CONTINUE
C
C READING VARIABLES FROM FILE 5 (INPUT.DAT) AND PREPARING
C HEADER OF FILE 12 (OUTPUT.OUT) WHERE WE HAVE A RUNNING SUMMARY.
C
WRITE(11,*) 'INPUT COMPLEX NAME.'
READ(5,*,ERR=101) NOMMOL
WRITE(11,*) 'COMPLEX:',NOMMOL
WRITE(11,*) 'INPUT NUMBER OF PROTONS:'
READ(5,*,ERR=101) NUPRO
IF (NUPRO .GT. 15) THEN
  WRITE(11,*) 'ERROR: NUMBER OF PROTONS > 15'
  GOTO 101
ENDIF
WRITE(11,*) 'NUMBER OF PROTONS: ',NUPRO
WRITE(12,*) 'COMPLEX: ',NOMMOL
WRITE(12,*) ''
WRITE(12,*) '** VALUES OF R. EXPLANATION:'
WRITE(12,*) '      1 - OK (MINIMUM)'
WRITE(12,*) '      2 - NOT OK. (MAXIMUM)'
WRITE(12,*) '      3 - NOT OK. (MINIMUM BUT DELTAC '
WRITE(12,*) '          SMALLER THAN EXPERIMENT)'
WRITE(12,*) '      9 - NOT OK. (NEGATIVES SOLUTIONS)'
WRITE(12,*) ''
C
WRITE(12,*) 'COMPLEX: ',NOMMOL
DO 1 J=1,NUPRO
  WRITE(11,*) 'INPUT PROTON NAME'

```

```

READ(5,*,ERR=101) NOMPROT
WRITE(11,*) 'PROTON...:',NOMPROT
WRITE(12,*) J;= PROTON:',NOMPROT
WRITE(11,*) 'INPUT NUMBER OF EXPERIMENTAL OBSERVATIONS OR '
1 'FI FOR FINNISHING'
READ(5,*,ERR=100) NOBS(J)
WRITE(11,*) 'OBSERVATIONS:',NOBS(J)
IF (NOBS(J) .GT. 100) THEN
  WRITE(11,*) 'ERROR: OBSERVATIONS BIGGER THAN 100'
  GOTO 100
ENDIF
WRITE(11,*) 'INPUT CONCENTRATION OF A, B AND SHIFTS'
WRITE(11,*) 'OBSERVATIONS:'
DELTAMIN(J)=1.D32
DELTAMAX(J)=0.D0
DO 11 I=1,NOBS(J)
  READ(5,*,ERR=101) A(J,I),B(J,I),DELTA(J,I)
  DELTAMIN(J)=MIN(DELTA(J,I),DELTAMIN(J))
  DELTAMAX(J)=MAX(DELTA(J,I),DELTAMAX(J))
  WRITE(11,*) A(J,I),B(J,I),DELTA(J,I)
11 CONTINUE
WRITE(11,*) 'MINIMUM DELTA:',DELTAMIN(J)
WRITE(11,*) 'MAXIMUM DELTA:',DELTAMAX(J)
C
C READING NOW KMIN AND KMAX BUT ONLY FOR FILES COMPATIBILITY
C WITH THE DRAWING PROGRAM: THEY ARE NO USED.
C
  WRITE(11,*) 'INPUT KMINIMUM AND KMAXIMUM'
  READ(5,*,ERR=101) KMIN,KMAX
  WRITE(11,*) 'K MINIMUM:',KMIN
  WRITE(11,*) 'K MAXIMUM:',KMAX
C
C READING NOW THE TOLERANCE AND THE MAXIMUM NUMBER OF ITERATIONS
C ALLOWED FOR THE PROGRAM
C
  WRITE(11,*) 'INPUT ALLOWED ERROR'
  READ(5,*,ERR=101) TOL
  WRITE(11,*) 'TOLERANCE:', TOL
  WRITE(11,*) 'INPUT MAXIMUM ITERATIONS'
  READ(5,*,ERR=101) NMAX
  WRITE(11,*) 'MAXIMUM ITERATIONS:',NMAX
1 CONTINUE
WRITE(12,*) ''
WRITE(12,*)
' J K INI. ASSOC. CONSTANT CALCD DELTA ERROR FUNC R'
WRITE(12,*)
'-----'
C
C NEWTON-RAPHSON METHOD (LOOP FOR EACH INITIAL K VALUE)
C
DO 50 COUNT=1,NKS
  K1=KINI(COUNT)
  WRITE(11,*) '++++++'
  WRITE(11,*) 'NEWTON-RAPHSON STARTS'
  WRITE(11,*) 'K INITIAL.....:',K1
  IERR=0
  DO 150 I=1,NMAX
    CALL CALCULF(A,B,DELTA,NOBS,K1,F,DF,DELTA(I),IERR,NUPRO)
    IF (IERR.EQ. 1) GOTO 2000

```

```

      IF (DABS(F) .LT. TOL) THEN
        WRITE(11,*) 'FINISHED (3)'
        WRITE(11,*) 'ITERATIONS',I
        CALL ESCRIURE(K1,F,NOBS,A,B,DELTA,NUPRO)
        GOTO 1000
      ENDIF
      IF (DABS(DF) .LT. TOL) THEN
        WRITE(11,*) 'DERIVATIVE SMALLER THAN TOLERANCE'
        WRITE(11,*) 'ITERATIONS',I
      ENDIF
      WRITE(11,*) 'ITERATION N-R: ',I
      WRITE(11,*) 'K1:',K1
      INCREM=-F/DF
      CALL ORIGINAL(A,B,DELTA,NOBS,K1,DELTAC,FORIG,IERR,NUPRO)
      WRITE(11,*) 'INCREMENT.....',INCREM
      WRITE(11,*) 'FUNCTION ORIG.....',FORIG
      WRITE(11,*) 'FUNCTION DERIV.....',F
      WRITE(11,*) 'SECOND DERIVATIVE.....',DF
      DO 1234 J=1,NUPRO
1234  WRITE(11,*) 'DELTA C (CALCULATED).....',DELTAC(J)
      IF (DABS(INCREM) .LT. TOL) THEN
        WRITE(11,*) 'INCREMENTS SMALLER THAN TOLERANCE'
        WRITE(11,*) 'ITERATIONS',I
      ENDIF
      K1=K1+INCREM
150  CONTINUE
      WRITE(11,*) 'MAXIMUM ITERATIONS EXCEEDED',NMAX
1000  CONTINUE
      IRES=1
      DO 324 J=1,NUPRO
        IF (DELTAC(J) .LT. DELTAMAX(J)) THEN
          WRITE(11,*) 'BE CAREFUL: DELTA FOUND'
          WRITE(11,*) 'IS SMALLER THAN MAXIMUM OBSERVED'
          IRES=3
        ENDIF
324  CONTINUE
      IF (DF .GT. 0.D0) THEN
C
C TESTING OF SOLUTIONS ON INITIAL EQUATION.
C
        WRITE(11,*) 'TESTING SOLUTIONS:'
        DO 250 J=1,NUPRO
          DO 250 I=1,NOBS(J)
            P=DELTA(J,I)/DELTAC(J)
            KTEOR=P/(1-P)/(B(J,I)-P*A(J,I))
            WRITE(11,*) 'P:',P,'K TEOR:',KTEOR
250  CONTINUE
          ELSE
            WRITE(11,*) 'IT WAS A MAXIMUM'
            IRES=2
          ENDIF
2000  CONTINUE
          IF (IERR.EQ.1) THEN
            WRITE(11,*) 'ERROR: THERE ARE NEGATIVES SOLUTIONS'
            K1=9.999999999999D0
            F=9.999999999999D0
            DF=9.999999999999D0
            DO 33 J=1,NUPRO
33  DELTAC(J)=9.999999999999D0

```

```

FORIG=9.999999999999D0
IRES=9
ENDIF
WRITE(11,*) ' END FOR A NEWTON-RAPHSON ITERATION+++++++'
WRITE(11,*) '+++++++'
DO 34 J=1,NUPRO
WRITE(12,888) J,' ',KINI(COUNT),K1,' ',DELTAC(J),' ',
I FORIG,' ',IRES
34 CONTINUE
WRITE(12,*) ''
50 CONTINUE
100 CONTINUE
WRITE(11,*) 'END OF JOB'
STOP
101 CONTINUE
WRITE(11,*) 'DATA ERROR'
STOP
888 FORMAT (1X,I2,A1,G7.1,E18.10,A1,E12.5,A1,E12.5,A2,I1)
END
C*****
C
C
C
C SUBROUTINES DESCRIPTION
C
C CALCULF:
C COMPUTATION OF THE FUNCTION TO BE SOLVED (F) AND OF ITS DERIVATIVE
C (DF). IT WORKS WITH ALL THE PROTONS AT ONCE (UP TO 15).
C IT IS A LOOP OF CALLS TO SUBROUTINE CALCULFC
C (WHICH SOLVES THE EQUATION FOR ONE PROTON).
C
C CALCULFC:
C COMPUTATION OF THE FUNCTION TO BE SOLVED (F) AND OF ITS DERIVATIVE
C (DF), WORKING WITH ONE PROTON.
C
C ORIGINAL:
C COMPUTATION OF THE FUNCTION TO BE SOLVED (FORIG).
C WOKS WITH ALL PROTONS SIMULTANEOUSLY (UP TO 15).
C THIS SUBROUTINE IS A LOOP OF CALLS TO SUBROUTINE
C ORIGINALC (WHICH WORKS WITH ONE SINGLE PROTON).
C
SUBROUTINE CALCULF(A,B,DELTA,NOBS,K,F,DF,DELTAC,IERR,NUPRO)
REAL*8 A(15,100),B(15,100),DELTA(15,100)
REAL*8 A2(100),B2(100),DELTA2(100)
REAL*8 K,F,DF,DELTAC(15)
REAL*8 F2,DF2,DELTAC2
INTEGER*4 NOBS(15),I,J,IERR,NUPRO
INTEGER*4 NOBS2
IERR=0
F=0.D0
DF=0.D0
C
C LOOP, FOR EACH PROTON, INCREASING THE FUNCTION
C
DO 10 J=1,NUPRO
NOBS2=NOBS(J)
DELTAC2=DELTAC(J)
F2=0.D0
DF2=0.D0

```

```

      DO 20 I=1,NOBS(J)
        B2(I)=B(J,I)
        A2(I)=A(J,I)
        DELTA2(I)=DELTA(J,I)
20    CONTINUE
      CALL CALCULFC(A2,B2,DELTA2,NOBS2,K,F2,DF2,DELTAC2,IERR)
      IF (IERR.EQ. 1) RETURN
      F=F+F2
      DF=DF+DF2
      DELTAC(J)=DELTAC2
10    CONTINUE
      RETURN
      END
C
C COMPUTATION OF THE FUNCTION FOR EACH OF THE PROTONS AND OF ITS
C DERIVATIVE (DF)
C
      SUBROUTINE CALCULFC(A,B,DELTA,NOBS,K,F,DF,DELTAC,IERR)
      REAL*8  A(100),B(100),DELTA(100)
      REAL*8  K,F,DF,DELTA2,DDELTA2
      REAL*8  L(100),S(100),DS(100),DL(100),Z(100)
      REAL*8  AUX0,AUX1,AUX2,AUX3,AUX4,AUX5,AUX6
      INTEGER*4 NOBS,I,IERR
      IERR=0
C
C COMPUTATION OF INTERMEDIATE VARIABLES: L(I),S(I),Z(I)
C AND OF THEIR PARTIAL DERIVATIVES
C
      DO 10 I=1,NOBS
        S(I)=A(I)+B(I)+(1.D0/K)
        Z(I)=(S(I)*S(I))-(4.D0*A(I)*B(I))
        IF ( Z(I) .LT. 0.D0) THEN
          WRITE(11,*) 'WARNING 1: FOR THIS INITIAL VALUE OF K '
          WRITE(11,*) 'THE PROGRAM DOES NOT CONVERG.'
          WRITE(11,*) 'PROCESS FOLLOWS FOR THE NEXT INITIAL K VALUE.'
          WRITE(11,*) 'Z(I): ',Z(I)
          WRITE(6,*) 'WARNING 1: FOR THIS INITIAL VALUE OF K '
          WRITE(6,*) 'THE PROGRAM DOES NOT CONVERG.'
          WRITE(6,*) 'PROCESS FOLLOWS FOR THE NEXT INITIAL K VALUE.'
          IERR=1
          RETURN
        ENDIF
        Z(I)=DSQRT(Z(I))
        L(I)=S(I)-Z(I)
        DS(I)=-1.D0/K/K
        DL(I)=L(I)/Z(I)/K/K
        AUX6=DS(I)*(1.D0-(S(I)/Z(I)))
        IF (DABS(AUX6-DL(I)) .GT. 1.D-6) THEN
          WRITE(11,*) 'WARNING 2: FOR THIS INITIAL VALUE OF K '
          WRITE(11,*) 'THE PROGRAM DOES NOT CONVERG.'
          WRITE(11,*) 'PROCESS FOLLOWS FOR THE NEXT INITIAL K VALUE.'
          WRITE(11,*) 'DL(I): ',DL(I)
          WRITE(11,*) 'AUX6.: ',AUX6
          WRITE(6,*) 'WARNING 2: FOR THIS INITIAL VALUE OF K '
          WRITE(6,*) 'THE PROGRAM DOES NOT CONVERG.'
          WRITE(6,*) 'PROCESS FOLLOWS FOR THE NEXT INITIAL K VALUE.'
          WRITE(6,*) 'DL(I): ',DL(I)
          WRITE(6,*) 'AUX6.: ',AUX6
          IERR=1
        ENDIF
      END DO

```

```

        RETURN
    ENDIF
10 CONTINUE
C
C COMPUTATION OF DELTAC
C
    AUX1=0.D0
    AUX2=0.D0
    DO 20 I=1,NOBS
        AUX1=AUX1+(L(I)*L(I)/A(I)/A(I)/2.D0)
        AUX2=AUX2+(DELTA(I)*L(I)/A(I))
    20 CONTINUE
    DELTAC=AUX2/AUX1
    AUX1=0.D0
    AUX2=0.D0
    DO 30 I=1,NOBS
        AUX1=AUX1+( L(I)*L(I)/ A(I)/ A(I) / 2.D0 / Z(I) )
        AUX2=AUX2+( DELTA(I)*L(I)/ Z(I) / A(I) )
    30 CONTINUE
C
C COMPUTATION OF THE FUNCTION AT THE POINT (F)
C
    F=(DELTAC*AUX1) - AUX2
C
C
    F=DELTAC * F
C
C COMPUTATION F DELTAC DERIVATIVE
C
    AUX1=0.D0
    AUX2=0.D0
    AUX3=0.D0
    AUX4=0.D0
    AUX5=0.D0
    DO 40 I=1,NOBS
        AUX1=AUX1+( DELTA(I) * DL(I) / A(I) )
        AUX2=AUX2+( L(I) * L(I) / A(I) / A(I) )
        AUX3=AUX3+( L(I) * DL(I) / A(I) / A(I) )
        AUX4=AUX4+( DELTA(I) * L(I) / A(I) )
    40 CONTINUE
    AUX5 = AUX2 * AUX2
    DDELTAC= 2.D0 * ( AUX1 * AUX2 - 2.D0 * AUX3 * AUX4 ) / AUX5
C
C COMPUTATION OF THE DERIVATIVE IN THE POINT
C
    AUX0=0.D0
    AUX1=0.D0
    AUX2=0.D0
    AUX3=0.D0
    AUX4=0.D0
    AUX5=0.D0
    DO 50 I=1,NOBS
        AUX1=AUX1+( L(I) * L(I) / 2.D0 / A(I) / A(I) / Z(I) )
        AUX0 = 2.D0 * L(I) * Z(I) * DL(I)
        AUX0 = AUX0 - ( L(I) * L(I) * ( DS(I) - DL(I) ) )
        AUX0 = AUX0 / 2.D0 / A(I) / A(I) / Z(I) / Z(I)
        AUX2=AUX2+AUX0
        AUX0 = DELTA(I) * DL(I) * Z(I)
        AUX0 = AUX0 - ( DELTA(I) * L(I) * ( DS(I) - DL(I) ) )
    50 CONTINUE

```

```

      AUX0 = AUX0 / Z(I) / Z(I) / A(I)
      AUX3 = AUX3 + AUX0
50  CONTINUE
      DF = ((DDELTA * AUX1) + (DELTA * AUX2) - AUX3)
C
C
      DF = DDELTA * F + DELTA * DF
      RETURN
      END
C
C COMPUTATION OF THE FUNCTION TO BE SOLVED (F) AND OF ITS
C DERIVATIVE (DF)
C
      SUBROUTINE ORIGINAL(A,B,DELTA,NOBS,K,DELTA,FORIG,IERR,NUPRO)
      REAL*8  A(15,100),B(15,100),DELTA(15,100)
      REAL*8  A2(100),B2(100),DELTA2(100)
      REAL*8  K,FORIG,DELTA(15),DELTA2
      REAL*8  FORIG2
      INTEGER*4 NOBS(15),I,J,IERR,NUPRO
      INTEGER*4 NOBS2
      IERR=0
C
C LOOP, FOR EACH PROTON, INCREASING THE FUNCTION
C
      FORIG=0.D0
      DO 10 J=1,NUPRO
        NOBS2=NOBS(J)
        DELTA2=DELTA(J)
        FORIG2=0.D0
        DO 20 I=1,NOBS(J)
          B2(I)=B(J,I)
          A2(I)=A(J,I)
          DELTA2(I)=DELTA(J,I)
20      CONTINUE
          CALL ORIGINAL2(A2,B2,DELTA2,NOBS2,K,DELTA2,FORIG2,IERR)
          IF (IERR.EQ. 1) RETURN
          FORIG=FORIG+FORIG2
          DELTA(J)=DELTA2
10      CONTINUE
      RETURN
      END
C
C COMPUTATION OF THE FUNCTION ORIGINAL
C
      SUBROUTINE ORIGINAL2(A,B,DELTA,NOBS,K,DELTA,FORIG,IERR)
      REAL*8  A(100),B(100),DELTA(100)
      REAL*8  K,DELTA
      REAL*8  L(100),S(100)
      REAL*8  AUX1,AUX2,AUX3,FORIG
      INTEGER*4 NOBS,I,IERR
      IERR=0
      DO 270 I=1,NOBS
        S(I)=A(I)+B(I)+(1.D0/K)
        AUX3=(S(I)*S(I))-(4.D0*A(I)*B(I))
        IF ( AUX3 .LT. 0.D0) THEN
          WRITE(11,*) 'WARNING 3: FOR THIS INITIAL VALUE OF K '
          WRITE(11,*) 'THE PROGRAM DOES NOT CONVERG.'
          WRITE(11,*) 'PROCESS FOLLOWS FOR THE NEXT INITIAL K VALUE.'
          WRITE(11,*) 'AUX3.: ',AUX3

```

```

WRITE(6,*) 'WARNING 3: FOR THIS INITIAL VALUE OF K '
WRITE(6,*) 'THE PROGRAM DOES NOT CONVERG.'
WRITE(6,*) 'PROCESS FOLLOWS FOR THE NEXT INITIAL K VALUE.'
WRITE(6,*) 'AUX3.:',AUX3
IERR=1
RETURN
ENDIF
L(I)=S(I)-DSQRT(AUX3)
270 CONTINUE
AUX1=0.D0
AUX2=0.D0
DO 80 I=1,NOBS
  AUX1=AUX1+(L(I)*L(I)/A(I)/A(I)/2.D0)
  AUX2=AUX2+(DELTA(I)*L(I)/A(I))
80 CONTINUE
DELTAC=AUX2/AUX1
AUX1=0.D0
AUX2=0.D0
DO 380 I=1,NOBS
  AUX1=AUX1+((DELTA(I)-(DELTAC*L(I)/2.D0/A(I))) ** 2)
380 CONTINUE
FORIG=AUX1
RETURN
END
C
C
C
SUBROUTINE ESCRIURE(K,ERR,NOBS,A,B,DELTA,NUPRO)
REAL*8 ERR,A(15,100),B(15,100),DELTA(15,100),K
REAL*8 FUNC,DELTAC(15),FORIG,DF
INTEGER*4 NOBS(15),I,J,IERR,NUPRO
WRITE(11,*) 'EQUILIBRIUM CONSTANT:',K
WRITE(11,*) 'ERROR:',ERR
DO 1001 J=1,NUPRO
  DELTAC(J)=0.D0
  WRITE(11,*) 'J: ',J
  DO 1001 I=1,NOBS(J)
1001 WRITE(11,*) A(J,I),B(J,I),DELTA(J,I),J,I
CALL CALCULF(A,B,DELTA,NOBS,K,FUNC,DF,DELTAC,IERR,NUPRO)
WRITE(11,*) 'FUNCTION: ',FUNC
DO 2001 J=1,NUPRO
2001 WRITE(11,*) 'DELTA C: ',DELTAC(J)
CALL ORIGINAL(A,B,DELTA,NOBS,K,DELTAC,FORIG,IERR,NUPRO)
WRITE(11,*) 'ORIGINAL FUNCTION FOR THE K FOUND: ',FORIG
RETURN
END

```


References

1. P. Job: *Ann. Chim.* **9**, 113 (1928).
2. G. Scatchard: *Ann. N.Y. Acad. Sci.* **51**, 660 (1949).
3. H.A. Benesi and J.H. Hildebrand: *J. Am. Chem. Soc.* **71**, 2703 (1949).
4. R.L. Scott: *Recl. Trav. Chim. Pays-Bas* **75**, 787 (1956).
5. (a) D.M. Himmelblau: *Applied Non-Linear Programming*, McGraw-Hill, New York, 1972; (b) M.A. Wolfe: *Numerical Methods for Unconstrained Optimization: an Introduction*, Van Nostrand Reinhold Co., New York, 1978; (c) R. Fletcher: *Practical Methods of Optimization*, John Wiley and Sons, New York, 1980.
6. D. Salvatierra, C. Jaime, A. Virgili, and F. Sánchez-Ferrando: *J. Org. Chem.* **61**, 9578 (1996).