*Analytica Chimica Acta*
Elsevier Publishing Company, Amsterdam
Printed in The Netherlands



# A COMPUTER PROGRAM FOR THE CALCULATION OF EQUILIBRIUM CONCENTRATIONS IN COMPLEX SYSTEMS

M. BOS and H. Q. J. MEERSHOEK
*Department of Chemical Technology, Twente University of Technology, Enschede (The Netherlands)*
(Received 24th January 1972)

The calculation of the curves of acid–base titrations in non-aqueous media is difficult, because the number of equilibria involved is often quite large. Depending on the nature of the solvent and the compounds titrated, many equilibria have to be taken into account.

A general and versatile computer program for calculating complex chemical equilibria is obviously very useful in studying titrations in non-aqueous media. Its utility, however, depends very much on the efficiency of the program and the method, as computer time quickly becomes prohibitive when the number of equilibria increases. Much work has already been done on writing computer programs for the calculation of equilibrium concentrations. A literature survey of this work is given by Zeleznik and Gordon[1]. A saving in computer time in comparison to programs published already, such as HALTAFALL[2,3] and EQUIBRAT[4], can be obtained by the use of the reaction extents as main variables combined with the stoichiometric coefficients in the equilibrium constant equations. A similar approach has also been suggested by Meissner *et al.*[5] and Bugaevski[6]. Because their methods were specially developed for hand computation, these authors solved the equations successively and iterated to the equilibrium concentrations by repeating this procedure. In a computer method it is better to solve the equations simultaneously. This saves computer time and makes it possible to write the equations in any order.

The preparation of the equations for the calculations from the chemical equilibria sometimes requires rather complicated "bookkeeping". If the programming language PL1 is used, this preparation can easily be automated.

## THEORY

In a mixture of compounds in which the following equilibria occur:

$$a_i\mathrm{A}+b_i\mathrm{B}+\ldots \overset{K_i}{\rightleftharpoons} p_i\mathrm{P}+q_i\mathrm{Q}+\ldots \qquad (1)$$

the equations for the equilibrium constants can be expressed as:

$$K_i\cdot[\mathrm{A}]^{a_i}[\mathrm{B}]^{b_i}\ldots=[\mathrm{P}]^{p_i}[\mathrm{Q}]^{q_i}\ldots \qquad (2)$$

where A, B, ... = reactants
P, Q, ... = products
$a, b, \ldots, p, q, \ldots$ = stoichiometric coefficients
$K$ = equilibrium constants
$i$ = number of reaction

If there are $N$ equilibria and $M$ compounds taking part in these equilibria, a matrix $V$ with $N$ rows and $M$ columns can be introduced, $V_{ij}$, containing the stoichiometric coefficient of compound $j$ in reaction $i$. $V_{ij}$ must be taken positive if compound $j$ is a reactant in reaction $i$, negative if compound $j$ is a product in reaction $i$, and zero if compound $j$ does not take part in reaction $i$.

The equilibrium constant equations (2) can now be written as:

$$K_i \prod_{j=1}^{M} c_j^{V_{ij}} = 1 \tag{3}$$

where $c_j$ represents the equilibrium concentration of compound $j$.

With introduction of the reaction extent, $x_i$, defined as the concentration decrease of a reactant with stoichiometric coefficient one caused by reaction $i$, the equilibrium concentrations can be expressed as follows:

$$c_j = c_{0j} - \sum_{i=1}^{N} V_{ij} x_i \tag{4}$$

where $c_{0j}$ denotes the initial concentration of compound $j$.

*Mathematical method*

The following set of N non-linear equations in the N unknown $x_i$ has to be solved:

$$f_i(\underline{x}) = \log K_i + \sum_{j=1}^{M} V_{ij} \log c_j = 0, \qquad i = 1, 2, \ldots N$$

with

$$c_j = c_{0j} - \sum_{i=1}^{N} V_{ij} x_i, \qquad j = 1, 2, \ldots M$$

($K_i$ is the given equilibrium constant of equation $i$; $c_{0j}$ is the given initial concentration of compound $j$; and the unknown $x_i$ is the reaction extent of equation $i$). $c_j$, depending upon $\underline{x}$, represents the required equilibrium concentration of compound $j$, and $V$ is the given matrix as described.

One may assume that:

$M \geqslant N$

rank of $V = M$

If this is not the case, the reaction equations should be dependent. To solve this set of equations the method of Newton is used. (N.B.: A vector is marked with an underlining.)

*Linearization*

Suppose a guess $\underline{\tilde{x}}$ (with corresponding $\underline{\tilde{c}} = \underline{c}_0 - V^T \underline{\tilde{x}}$) is available for the unknown $\underline{x}$. Developing $\underline{f}(\underline{x})$ around $\underline{\tilde{x}}$:

$$\underline{f}(\underline{\tilde{x}} + \mathrm{d}\underline{x}) = \underline{f}(\underline{\tilde{x}}) + Z\,\mathrm{d}\underline{x} + \ldots$$

where $Z$ is a $N$ by $N$ matrix.

$$Z_{ij} = \frac{\delta f_i}{\delta x_j} = \sum_{l=1}^{M} \frac{\delta f_i}{\delta c_l} \cdot \frac{\delta c_l}{\delta x_j} = -\sum_{l=1}^{M} V_{il} \frac{1}{c_l} \cdot V_{jl}$$

or $Z = -VC^{-1}V^T$ with $C$ a diagonal matrix ($C_{ii} = \tilde{c}_i$).

The linearized set of equations is:

$$VC^{-1}V^T \mathrm{d}\underline{x} = \underline{f}(\tilde{\underline{x}})$$

It is simple to verify that the matrix $VC^{-1}V^T$ is symmetric and positive definite for positive concentrations.

For the expression

$$(VC^{-1}V^T\underline{u}, \underline{u}) = (C^{-1}V^T\underline{u}, V^T u) > 0$$

is valid for any $\underline{u} \neq \underline{0}$; thus the matrix $VC^{-1}V^T$ is regular in the whole region $\underline{c} > \underline{0}$, and so there exists at most one solution.

During the iterative process the concentrations should remain positive. Instead of the correction $\mathrm{d}\underline{x}$ (with the corresponding correction $\mathrm{d}\underline{c} = -V^T\mathrm{d}\underline{x}$) the correction $\alpha\mathrm{d}\underline{x}$ is made.

Initially, for $\alpha$, the maximal value ($\alpha_0$) is chosen with

$$\alpha \leqslant 1 \quad \text{and} \quad \tilde{\underline{c}} + \alpha \mathrm{d}\underline{c} \geqslant p_0 \tilde{\underline{c}}$$

with $p_0$ a constant (*e.g.* $p_0 = 0.001$).

For sufficiently small positive $\alpha$

$$||\underline{f}(\tilde{\underline{x}} + \alpha \mathrm{d}\underline{x})|| \leqslant ||\underline{f}(\tilde{\underline{x}})||$$

assuming $\underline{f}(\tilde{\underline{x}}) \neq \underline{0}$.

$\alpha_0$ is halved as long as this equation is not satisfied. From an initial guess $\underline{x}^{(0)}$, a row of vectors $\underline{x}^{(k)}$ with $k = 1, 2, \ldots$ is obtained. The corresponding row of vectors $\underline{f}^{(k)} = \underline{f}(\underline{x}^{(k)})$ converges monotonously to zero. Consequently the row $\underline{x}^{(k)}$ converges to the solution. In the vicinity of the solution, the convergence is of the second order, *i.e.* in an iterative step $k$ the norm of the difference between the iterand $\underline{x}^{(k-1)}$ and the solution $\underline{x}^{(\infty)}$ is squared or:

$$||\underline{x}^{(k)} - \underline{x}^{(\infty)}|| \sim ||\underline{x}^{(k-1)} - \underline{x}^{(\infty)}||^2$$

*Initial guess*

The only requirement for the initial guess $\underline{x}^{(0)}$ is that the corresponding concentrations $\underline{c}^{(0)} = \underline{c}_0 - V^T\underline{x}^{(0)}$ satisfy the relation $\underline{c}^{(0)} > \underline{0}$. At first $\underline{x}^{(0)}$ is set zero, and consequently $\tilde{\underline{c}}^{(0)} = \underline{c}_0$.

One can simply assume that each row of $V$ contains both positive and negative coefficients. Then an equation is sought of which the product of the terms of the left side $= 0$ and the product of the terms of the right side $\neq 0$, or the reverse.

Suppose equation $i$ is found.

Put:
$$\alpha_i = \frac{\min\limits_{j;V_{ij}>0} c_j}{\max\limits_{j;V_{ij}>0} V_{ij}}$$

$$\beta_i = \frac{\min\limits_{j;V_{ij}<0} c_j}{\max\limits_{j;V_{ij}<0} |V_{ij}|}$$

Either $\alpha_i = 0$ and $\beta_i \neq 0$, or $\alpha_i \neq 0$ and $\beta_i = 0$. By putting $x_i^{(0)} = \frac{1}{2}(\alpha_i - \beta_i)$ the concentrations are changed in the right direction. At present all the concentrations which have been met in equation $i$, are positive. The process is repeated until all the concentrations are positive. This is the case after maximal $N$ steps.

*Numerical aspects*

It has been shown that the matrix $Z = VC^{-1}V^T$ is regular in the whole region $\underline{c} > \underline{0}$.

From a numerical point of view the matrix $Z$ can be singular in the case of loss of significance. With the IBM 360 it is advisable to compute in double length (15 digits). Only in very extreme situations can an unacceptable loss of significance occur.

In solving the linearized set of equations $Z\mathrm{d}\underline{x} = \underline{f}$, use is made of the particular properties of the matrix $Z$, *i.e.* symmetric and positive definite. The procedures used, DETSYM and SØLSYM, are very exact.

*Precipitates*

If compound $j$ forms a precipitate, then this compound does not take part in the equilibrium equations, or in other words with respect to the iteration column $j$ of matrix $V$ is considered zero. With respect to the initial guess, the original matrix $V$ is considered and precipitate formation is not taken into account. If precipitates are added in the starting mixture, the "formal" concentration should be given as initial concentrations for these compounds. Consequently all the concentrations, calculated from the initial guess $x^{(0)}$, are positive, including the precipitates. Finally the concentrations are calculated from the reaction extent $\underline{x}$ and the original matrix $V$, precipitate formation being neglected. In this case it is possible that a concentration, corresponding to a precipitate, is negative; a warning is then printed. This means, in fact, that precipitation does not occur or that a compound which is added in the initial mixture dissolves completely and the corresponding equilibrium should be omitted from the list of equilibrium equations that describe the system.

## PROGRAM DESCRIPTION

The program used is shown in the Appendix.

*Input*

The data are put on cards, consecutively, apart from blanks. With regard to the input formats, the following rules should be respected.

*Equations.* The "equations" have to be closed by '*' and separated by ';'. It is advisable to take a new card for each equation.

An "equation" consists of: "left-side", '=', and "right side".

A (left or right) "side" consists of one or more "terms", separated by '+'.

A "term" is the combination of a stoichiometric coefficient and the chemical formula of a compound in the chemical equilibrium. It thus consists of a "coefficient" (a digit from 1 to 9; if the coefficient = 1, it can be omitted), and the "name of the symbol", representing a compound. In the case of a precipitate or solvent, the name is followed by the '%'-sign. A "term" can be preceded and followed by blanks. The "name of the symbol" exists of letters (A–Z), digits (1–9) and ')' or '(', but it must not begin with a digit.

*Initial concentrations.* The "names of the symbols" of the compounds which have an initial concentration (suppose the number = MO) are closed by ';' (or closed by blanks followed by ';') and separated by one or more blanks.

MO numbers, *i.e.* the initial concentrations, in the right order, are closed and separated by blanks and/or ','.

*Equilibrium constants.* N numbers, being the equilibrium constants, are closed and separated by blanks and/or ','.

*Output*

The input about chemical equations and formulae is printed according to exactly the same format as the actual input, and the numerical data according to a slightly modified format.

The moment an error has been detected, an error message is printed (with an error-number); and the process is stopped.

The results of the program are printed according to the format:

line 1 to $M$:
per line $j$: name of symbol of compound $j$
initial concentration $= c_0\ (j)$
equilibrium concentration $= c(j)$
line 1 to $N$:
per line $i$: equilibrium constant $= k(i)$
reaction extent $= x(i)$

If an equilibrium concentration is negative, which is only possible for precipitates, a warning is printed.

*List of the most important symbols*

| | |
|---|---|
| $N$ | number of equations. |
| $M$ | number of compounds. |
| $V$ | $N$ by $M$ matrix, see description of methods, declared as $V$ (NMAX, MMAX). |
| $VV$ | the same matrix as $V$, but with the correct upper limits. |
| $VP$ | the same matrix as $VV$, but with the difference: $VP(I, J) = 0$ if compound $j$ is a precipitate. |
| $c_0$ | vector containing the initial concentrations. |
| $MO$ | number of initial concentrations different from zero. |
| $c$ | vector containing the equilibrium concentrations. |
| TSYMB | vector containing the names of symbols representing the compounds. |
| TPREC | vector; if TPREC($j$) = 1 then compound $j$ is a precipitate, otherwise TPREC($j$) = 0. |
| LØGK | vector containing the logarithms of the equilibrium constants. |
| $x$ | vector containing the reaction extents. |
| ØRDECO | vector; in this vector the order of the read-in initial concentrations is preserved. |
| $Z$ | the ($N$ by $N$) matrix of the linearized set of equations. |

*Program modules*

The program consists of the following modules:

a. Read-in the input
entry: data on cards
exit: N,M,$V$,$c_0$, TSYMB, TPREC, LØGK
b. Initial guess of $c$
entry: N,M,$V$,$c_0$
exit: $x$,$c$
c. Iteration
entry: N,M,$V$,$c$,$x$, LØGK
exit: $c$,$x$
Finally the results are printed.

*Detailed program description*

*Read-in the reaction equations.* The procedure READC reads in a character and prints it. If NCHAR (=the number of read-in characters) is a multiple of 80 (=the number of columns on a card) a line is printed. According to the definitions of the concepts "Name of Symbols", "Term" and "Side", there are the procedures NAME, TERM and SIDE to read in these quantities.

At the entrance of NAME the first character has been read-in already, and has been put in CHAR. All the three procedures read-in upto and including the first character which does not belong to the definition; at the exit this character has been placed in CHAR. While reading, the syntactical rules are checked thoroughly.

In the procedure SIDE the matrix $V$ and the vectors TSYMB and TPREC are filled.

*Read-in the initial concentrations.* The names of the symbols representing the compounds with initial concentrations are read-in. The order is preserved in ORDECO. Finally the initial concentrations are read-in and stored in the right place in $c_0$.

*Read-in the equilibrium constants.* The logarithms of these values are preserved in the array LØGK.

*Initial guess of $x$ (and $c$).* Initially $\underline{x}^{(0)}$ is set zero. The equations are treated (one or more times) from number 1 up to and including number $N$. If all the concentrations, which have been found in an equation $i$, are positive, then the next equation is processed. All concentrations of equation $i$ are certainly positive if $x^{(0)}(i) \neq 0$. If at least one concentration on both the left side and the right side of equation $i$ is zero, then the $x^{(0)}(i)$ cannot be changed; processing has to be delayed until a further round and the variable READY is set zero. If in a round no $x(i)$ has been changed (in that case the variable FIRST is still zero), while only some concentrations are equal to zero, the set of reaction equations or the list of starting materials, is wrong: some equations do not take part in the equilibrium.

*Iteration.* Initially a test is made to ensure that the rank of $V = N$. This is the case, when the matrix $VV^T$ (taking into account precipitates) is not singular. The process is then carried out according to the mathematical description. For $||f||$ the maximum norm is used. If for all $j$ the relative correction of $c_j <$ tol $c$, then the process is ended. After ITMAX (= 50) iterations, the process is stopped and an error message is printed.

*List of errors*
*code*
111 number of characters of a symbol > LMAX

112 number of compounds > MMAX;
113 number of equations > NMAX;
12 a compound appears in the equilibria in a soluble as well as a precipitated form;
121 '=' sign is missing in an equation;
122 the set of equations is not closed by the '*' sign;
14 a compound appears in the same equation for the second time;
15 a compound appears in the list of initial concentrations for the second time;
16 incorrect name of compound in the list of the initial concentrations;
17 an initial concentration is negative;
18 an equilibrium constant is negative;
21 in the procedure GUESS it appears that the left or right side in an equation is missing;
22 in the procedure GUESS it appears that some equations do not have a role; it is likely that there are too few starting materials;
31 the reaction equations are dependent;
32 the matrix $Z = VC^{-1}V^{T}$ proves to be singular by loss of significance;
33 after ITMAX iterations, the process has not been ended.

*Restrictions and possibilities for modifications*

The number of characters of a symbol, the number of equations and the number of compounds is restricted to 20. Each of these maxima can be easily changed (by modifying LMAX, NMAX and MMAX).

The maximum number of iterations (ITMAX) and the relative tolerance of CTØLC can also be easily modified.

If for one system many computations have to be made, *e.g.* with different initial concentrations and/or with different equilibrium constants, a slight modification of the program must be made. The final reaction extents of one calculation can be used to calculate the initial guess of the concentrations of the next by the use of eqn. (4).

When in the calculation of titration curves the dilution effect must be taken into account, it is necessary to correct the reaction extents by the dilution factor before using them in the calculation of the initial guess of the concentrations in the next point of the titration curve.

It is also necessary to prevent occurrence of zero concentrations in the initial guess. This can be done by replacing these zero concentrations by a very small concentration.

*Some results*

Tanaka and Nakagawa[7] calculated the curves for the titration of N,N-diethylaniline with perchloric acid in acetic acid which contained 0.5 *M* water.
With the program EQUIL the same titration curves were calculated, by means of the equilibria:

| | |
|---|---|
| $HX = H^{+} + X^{-}$; | $K_{HX} = 1.4 \cdot 10^{-5}$ |
| $B + HAc = BH^{+} + Ac^{-}$; | $K_{B} = 1.7 \cdot 10^{-6}$ |
| $BH^{+}X^{-} = BH^{+} + X^{-}$; | $K_{d}^{BH^{+}X^{-}} = 1.7 \cdot 10^{-6}$ |
| $HAc = H^{+} + Ac^{-}$; | $K_{s} = 3.5 \cdot 10^{-15}$ |
| $H_2O + HAc = H_3O^{+} + Ac^{-}$; | $K_{H_2O} = 8.4 \cdot 10^{-11}$ |

TABLE I

EQUILIBRIUM CONCENTRATIONS IN THE TITRATION OF 0.01 *M* N,N-DIETHYLANILINE WITH PERCHLORIC ACID IN ACETIC ACID SOLVENT
(0.5 *M* water present)

| *Compound* | *Titration degree* | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 0.90 | | 0.95 | | 1.00 | | 1.05 | | 1.10 | |
| | *Initial conc.* | *Equil. conc.* | *Initial conc.* | *Equil. conc.* | *Initial conc.* | *Equil. conc.* | *Initial conc.* | *Equil. conc.* | *Initial conc.* | *Equil. conc.* |
| HX | $8.92 \cdot 10^{-3}$ | $2.27 \cdot 10^{-9}$ | $9.41 \cdot 10^{-3}$ | $4.75 \cdot 10^{-9}$ | $9.90 \cdot 10^{-3}$ | $4.87 \cdot 10^{-8}$ | $1.04 \cdot 10^{-2}$ | $1.61 \cdot 10^{-6}$ | $1.09 \cdot 10^{-2}$ | $6.08 \cdot 10^{-6}$ |
| $H^+$ | 0.00 | $2.67 \cdot 10^{-10}$ | 0.00 | $5.23 \cdot 10^{-10}$ | 0.00 | $4.28 \cdot 10^{-9}$ | 0.00 | $4.13 \cdot 10^{-8}$ | 0.00 | $8.21 \cdot 10^{-8}$ |
| $X^-$ | 0.00 | $1.15 \cdot 10^{-4}$ | 0.00 | $1.22 \cdot 10^{-4}$ | 0.00 | $1.53 \cdot 10^{-4}$ | 0.00 | $5.25 \cdot 10^{-4}$ | 0.00 | $1.00 \cdot 10^{-3}$ |
| B | $9.92 \cdot 10^{-3}$ | $9.81 \cdot 10^{-4}$ | $9.91 \cdot 10^{-3}$ | $4.95 \cdot 10^{-4}$ | $9.90 \cdot 10^{-3}$ | $5.06 \cdot 10^{-5}$ | $9.90 \cdot 10^{-3}$ | $1.55 \cdot 10^{-6}$ | $9.89 \cdot 10^{-3}$ | $4.11 \cdot 10^{-7}$ |
| HAc | 16.50 | 16.50 | 16.50 | 16.50 | 16.50 | 16.50 | 16.50 | 16.50 | 16.50 | 16.50 |
| $BH^+$ | 0.00 | $1.24 \cdot 10^{-4}$ | 0.00 | $1.23 \cdot 10^{-4}$ | 0.00 | $1.03 \cdot 10^{-4}$ | 0.00 | $3.04 \cdot 10^{-5}$ | 0.00 | $1.60 \cdot 10^{-5}$ |
| $Ac^-$ | 0.00 | $1.31 \cdot 10^{-5}$ | 0.00 | $6.69 \cdot 10^{-6}$ | 0.00 | $8.17 \cdot 10^{-7}$ | 0.00 | $8.48 \cdot 10^{-8}$ | 0.00 | $4.26 \cdot 10^{-8}$ |
| $BH^+X^-$ | 0.00 | $8.80 \cdot 10^{-3}$ | 0.00 | $9.29 \cdot 10^{-3}$ | 0.00 | $9.75 \cdot 10^{-3}$ | 0.00 | $9.86 \cdot 10^{-3}$ | 0.00 | $9.87 \cdot 10^{-3}$ |
| $H_2O$ | 0.500 | 0.500 | 0.500 | 0.500 | 0.500 | 0.500 | 0.500 | 0.500 | 0.500 | 0.500 |
| $H_3O^+$ | 0.00 | $3.21 \cdot 10^{-6}$ | 0.00 | $6.28 \cdot 10^{-6}$ | 0.00 | $5.14 \cdot 10^{-5}$ | 0.00 | $4.95 \cdot 10^{-4}$ | 0.00 | $9.83 \cdot 10^{-4}$ |

TABLE II

TITRATION OF 0.01 *M* DIETHYLANILINE WITH PERCHLORIC ACID IN ACETIC ACID (0.5 *M* water present)

| *Degree of titration* | *pH* | | $p[H^+ClO_4^-]$ | |
|---|---|---|---|---|
| | *Tanaka* | *Present calcns.* | *Tanaka* | *Present calcns.* |
| 0.90 | 9.6 | 9.57 | 8.5 | 8.64 |
| 0.95 | 9.3 | 9.28 | 8.2 | 8.32 |
| 1.00 | 6.8 | 8.37 | 5.7 | 7.31 |
| 1.05 | 5.0 | 7.38 | 4.1 | 5.80 |
| 1.10 | 4.7 | 7.09 | 3.4 | 5.21 |

The results of the calculations are given in Table I, and a comparison with the results of Tanaka and Nakagawa[7] is given in Table II.

On the basic side, there is good agreement. However, at and after the equivalence point there are differences, because the assumption of Tanaka and Nakagawa[7] that $[X^-]$ can be neglected is not valid for perchloric acid. In fact $[X^-]$ is about 10% of the total acid concentration at 1:1 neutralization.

The authors wish to thank Miss A. L. Dekkers for preparing the manuscript.

SUMMARY

A general computer program has been developed for the calculation of equilibrium concentrations in complex systems. The use of the reaction extents as master variables in the equilibrium constant equations yields a considerable saving in computer time in comparison to existing programs. The setting up of the equations for the calculations from the chemical equilibria can be automated.

RÉSUMÉ

Un programme général d'ordinateur est proposé pour le calcul des concentrations d'équilibre dans des systèmes complexes. L'établissement des équations pour les calculs des équilibres chimiques peut être automatisé.

ZUSAMMENFASSUNG

Es wurde ein allgemeines Rechenprogramm für die Berechnung von Gleichgewichtskonzentrationen in Komplexsystemen entwickelt. Die Verwendung des Reaktionsausmasses als Hauptvariable in den Gleichungen der Gleichgewichtskonstante führt zu einer beträchtlichen Einsparung an Rechenzeit im Vergleich zu bisher verwendeten Programmen. Die Aufstellung der Gleichungen für die Berechnungen aus den chemischen Gleichgewichten kann automatisiert werden.

APPENDIX

SOURCE LISTING.

```
EQUIL: PROCEDURE OPTIONS(MAIN);
DETSYM:PROCEDURE(A,N,MINPIV);
       DCL (A(*,*),MINPIV) FLOAT BIN(53),
           N FIXED BIN,
           (R,S) FLOAT BIN (53),
           (K,L,J) FIXED BIN;
       DO K=1 TO N;
         R=A(K,K);
         DO L=1 TO K-1;
           R=R-A(L,K)*A(L,K);
         END;
         IF K=1 THEN MINPIV=R;
         MINPIV=MIN(MINPIV,R);
         IF R<= 0 THEN GOTO LSYM;
         A(K,K),R=SQRT(R);
         DO J=K+1 TO N;
           S=A(K,J);
           DO L=1 TO K-1;
             S=S-A(L,J)*A(L,K);
           END;
           A(K,J)=S/R;
         END;
       END;
LSYM:
END DETSYM;
SOLSYM:PROCEDURE (A,N,B);
       DCL (A(*,*),B(*)) FLOAT BIN (53),
           N FIXED BIN,
           S FLOAT BIN (53),
           (I,K) FIXED BIN;
       DO I=1 TO N;
         S=B(I);
         DO K=1 TO I-1;
           S=S-A(K,I)*B(K);
         END;
         B(I)=S/A(I,I);
       END;
       DO I=N TO 1 BY -1;
         S=B(I);
         DO K= I+1 TO N;
           S=S-A(I,K)*B(K);
         END;
         B(I)=S/A(I,I);
       END;
END SOLSYM;
GUESS: PROCEDURE(N,M,V,CO,X,C);
       DCL (V(N,M),X(N),(CO,C) (M)) FLOAT BIN(53),
           (N,M) FIXED BIN;
       DCL (COSUM,AT,BT,AN,BN,A,B) FLOAT BIN(53),
           (I,J,FIRST,READY) FIXED BIN;
       X=0;
       C=CO;
       COSUM=SUM(CO);
LB1:   FIRST=0 ; READY=1;
```

```
        DO I=1 TO N;
          IF X(I) ¬=0 THEN GOTO LB2;
          AT,BT=CCSUM;
          AN,BN=0;
          DO J=1 TO M;
            IF V(I,J)>0 THEN DO; AT=MIN(AT,C(J));
                                 AN=MAX(AN,V(I,J));
                             END;
            IF V(I,J)<0 THEN DO; BT=MIN(BT,C(J));
                                 BN=MAX(BN,-V(I,J));
                             END;
          END;
          IF AN=0 | BN=0 THEN CALL FAULT(21);
          A=AT/AN; B=BT/BN;
          IF A=0 & B=0 THEN DO; READY=0;
                                GOTO LB2;
                            END;
          IF A¬=0 & B¬=0 THEN GOTO LB2;
          FIRST=1;
          X(I)=0.5*(A-B);
          C=C-X(I)*V(I,*);
LB2:    END;
        IF READY=1 THEN RETURN;
        IF FIRST=1 THEN GOTO LB1;
        CALL FAULT(22);
        RETURN;
END GUESS;
ITER:   PROCEDURE(N,M,V,C,X,LOGK);
        DCL (V(N,M),C(M),(X,LOGK) (N)) FLOAT BIN(53),
            (N,M) FIXED BIN;
        DCL (W(N,M),Z(N,N),(F,DX) (N),(DC,LOGC) (M),
             TOLC,EPSC,P0,MINPIV,FMAX0,FMAX1,WR,ALFA) FLOAT BIN(53),
            (I,J,IT,ITMAX) FIXED BIN;
        ITMAX=150;
        TOLC=1.0E-5;
        P0=1.0E-3;
        DO I=1 TO N;
          DO J=1 TO I;
            Z(I,J),Z(J,I)=SUM(V(I,*)*V(J,*));
          END;
        END;
        CALL DETSYM(Z,N,MINPIV);
        WR=Z(1,1);
        DO I=2 TO N;
          WR=WR*Z(I,I);
        END;
        IF MINPIV <=0 | WR < 0.5 THEN CALL FAULT(31);
        FMAX0=0;
        DO J=1 TO M;
          LOGC(J)=LOG(C(J));
        END;
        DO I=1 TO N;
          F(I)=LOGK(I)+SUM(V(I,*)*LOGC);
          FMAX0=MAX(FMAX0,ABS(F(I)));
        END;
        DO IT=1 TO ITMAX;
          DO J=1 TO M;
            W(*,J)=V(*,J)/C(J);
          END;
          DO I=1 TO N;
```

```
            DO J=1 TO I;
               Z(I,J),Z(J,I)=SUM(W(I,*)*V(J,*));
            END;
         END;
         CALL DETSYM(Z,N,MINPIV);
         IF MINPIV <= 0 THEN CALL FAULT(32);
         CALL SOLSYM(Z,N,F);
         DX=F;
         ALFA=1.0E0;
         EPSC=0;
         DO J=1 TO M;
            DC(J)=-SUM(V(*,J)*DX);
            WR=DC(J)/C(J);
            EPSC=MAX(EPSC,ABS(WR));
            IF 1.0E0+ALFA*WR < P0 THEN ALFA=(P0-1.0E0)/WR;
         END;
         IF EPSC < TOLC THEN RETURN;
LC:      DO J=1 TO M;
            LOGC(J)=LOG(C(J)+ALFA*DC(J));
         END;
         FMAX1=0;
         DO I=1 TO N;
            F(I)=LOGK(I)+SUM(V(I,*)*LOGC);
            FMAX1=MAX(FMAX1,ABS(F(I)));
         END;
         IF FMAX1 > FMAX0 THEN DO; ALFA=0.5*ALFA;
                                   GOTO LC;
                                END;
         FMAX0=FMAX1;
         X=X+ALFA*DX;
         C=C+ALFA*DC;
      END;
   PUT LIST(X);
      CALL FAULT(33);
      RETURN;
END ITER;
      DCL FAULT ENTRY (FIXED BIN);
FAULT: PROCEDURE(ERROR);
      DCL ERROR FIXED BIN;
      PUT SKIP LIST ('ERROR=',ERROR);
      GOTO LAE;
END FAULT;
      DCL (NMAX,MMAX,LMAX) FIXED BIN;
      NMAX=20;
      MMAX=20;
      LMAX=10;
BLOCK1:
BEGIN;
READC: PROCEDURE;
      /* NON-LOCAL VARIABLES  CHAR,NCHAR; */
      GET EDIT (CHAR) (A(1));
      PUT EDIT (CHAR) (A(1));
      NCHAR=NCHAR+1;
      IF NCHAR=80 THEN DO; NCHAR=0;
                           PUT SKIP;
                        END;
      RETURN;
END READC;
NAME:  PROCEDURE;
      DCL L FIXED BIN;
```

```
        /* NON-LOCAL VARIABLES: ALP,ALPNUM,LMAX,CHAR,SYMB,PREC; */
        IF INDEX(ALP,CHAR)=0 THEN CALL FAULT(10);
        SYMB=CHAR;
        L=1;
        PREC=0;
LA11:   CALL READC;
        IF INDEX(ALPNUM,CHAR)¬=0
           THEN DO; L=L+1;
                    IF L>LMAX THEN CALL FAULT(111);
                    SYMB=SYMB||CHAR;
                    GOTO LA11;
                END;
        IF CHAR = '%' THEN DO; PREC=1;
                               CALL READC;
                           END;
        RETURN;
END NAME;
TERM:   PROCEDURE;
        DCL I FIXED BIN;
        /* NON-LOCAL VARIABLES:     CHAR,COEFF,NUM; */
LA21:   CALL READC;
        IF CHAR = ' ' THEN GOTO LA21;
        COEFF=1;
        I=INDEX(NUM,CHAR);
        IF I ¬=0 THEN DO; COEFF=I;
                          CALL READC;
                      END;
        CALL NAME;
LA22:   IF CHAR= ' ' THEN DO; CALL READC;
                              GOTO LA22;
                          END;
        RETURN;
END TERM;
SIDE:   PROCEDURE;
        DCL (J,JJ) FIXED BIN;
        /* NON-LOCAL VARIABLES:
           MMAX,M,N,SYMB,PREC,TSYMB,TPREC,SIGNUM,COEFF,V;   */

LA31:   CALL TERM;
        DO J=1 TO M;
          IF SYMB=TSYMB(J) THEN
            DO; JJ=J;
                IF TPREC(J) ¬= PREC THEN CALL FAULT(12);
                GOTO LA32;
            END;
        END;
        M,JJ=M+1;
        IF M>MMAX THEN CALL FAULT(112);
        TSYMB(M)=SYMB;
        TPREC(M)=PREC;
LA32:   IF V(N,JJ) ¬=0 THEN CALL FAULT(14);
        V(N,JJ)=SIGNUM*COEFF;
        IF CHAR='+' THEN GOTO LA31;
        RETURN;
END SIDE;
        DCL (TSYMB(MMAX),SYMB) CHAR(LMAX) VAR,
            ALP CHAR(28),NUM CHAR(9),ALPNUM CHAR(37),CHAR CHAR(1),
            (V(NMAX,MMAX),SIGNUM,COEFF) FLOAT BIN(53),
            ((TPREC,ORDECO) (MMAX),
             PREC,NCHAR,N,M,I,J,M0) FIXED BIN;
```

```
        ALP='ABCDEFGHIJKLMNOPQRSTUVWXYZ()';
        NUM='123456789';
        ALFNUM=ALP||NUM;
        V=0;
        M=0;
        NCHAR=0;
        N=1;
        PUT LIST ('EQUIL. EQUATIONS');
        PUT SKIP(2);
LA41:   SIGNUM=1;
        CALL SIDE;
        IF CHAR ¬= '=' THEN CALL FAULT(121);
        SIGNUM=-1;
        CALL SIDE;
        IF CHAR = ';' THEN
          DO; N=N+1;
              IF N>NMAX THEN CALL FAULT(113);
              GOTO LA41;
          END;
        IF CHAR ¬= '*' THEN CALL FAULT(122);
BLOCK2:
BEGIN;
        DCL VV(N,M) FLOAT BIN(53) DEF V,
            (VP(N,M),(X,KI,LOGK) (N),(CO,C) (M)) FLOAT BIN(53);
        PUT SKIP(2) LIST ('INITIAL CONC.');
        IF NCHAR=0 THEN PUT SKIP(2);
                   ELSE PUT SKIP EDIT (' ') (COLUMN(NCHAR),A(1));
        CO=0;
        MO=0;
LA42:   CALL READC;
        IF CHAR = ' ' THEN GOTO LA42;
LA43:   CALL NAME;
        DO J=1 TO M;
          IF TSYMB(J)=SYMB THEN
            DO; IF CO(J) ¬=0 THEN CALL FAULT(15);
                CO(J)=1;
                MO=MO+1;
                ORDECO(MO)=J;
                GOTO LA44;
            END;
        END;
        CALL FAULT(16);
LA44:   IF CHAR = ' ' THEN DO; CALL READC;
                               GOTO LA44;
                           END;
        IF CHAR ¬= ';' THEN GOTO LA43;
        PUT SKIP(2);
        DO I=1 TO MO;
          J=ORDECO(I);
          GET LIST(CO(J));
          PUT EDIT (CO(J)) (X(2),E(11,5,5));
          IF CO(J) <=0 THEN CALL FAULT(17);
        END;
        PUT SKIP(2) LIST ('EQUIL. CONSTANTS');
        PUT SKIP(2);
        DO I=1 TO N;
          GET LIST (KI(I));
          PUT EDIT (KI(I)) (X(2),E(11,5,5));
          IF KI(I) <=0 THEN CALL FAULT(18);
          LOGK(I)=LOG(KI(I));
        END;
```

```
        CALL GUESS(N,M,VV,CO,X,C);
        VP=VV;
        DO J=1 TO M;
          IF TPREC(J)=1 THEN VP(*,J)=0;
        END;
        CALL ITER(N,M,VP,C,X,LOGK);
        PUT SKIP(8) EDIT ('NAME') (X(4),A(4));
        PUT EDIT ('INIT. CONC.') (X(LMAX),A(11));
        PUT EDIT ('EQUIL. CONC.') (X(4),A(12));
        PUT EDIT ('EQUIL. CONST.') (COLUMN(LMAX+61),A(13));
        PUT EDIT ('REACTION EXTENT') (X(5),A(15));
        PUT SKIP;
        DO J=1 TO MAX(N,M);
          PUT SKIP;
          IF J > M THEN GOTO LA46;
          IF TPREC(J)=0 THEN GOTO LA45;
          C(J)=CO(J)-SUM(VV(*,J)*X);
LA45:
          PUT EDIT (J,TSYMB(J),CO(J),C(J))
                   (F(2),X(2),A(LMAX),(2) (X(4),E(11,5,5)));
          IF C(J) <=0 THEN PUT EDIT (' NEGATIVE !') (A(11));
LA46:     IF J > N THEN GOTO LA47;
          PUT EDIT (J,KI(J),X(J))
                   (COLUMN(LMAX+56),F(2),X(4),E(11,5,5),
                                              X(4),E(21,15,15));
LA47:
        END;
END BLOCK2;
END BLOCK1;
LAE:
END EQUIL ;
```

## REFERENCES


1 F. J. Zeleznik and S. Gordon, *Ind. Eng. Chem.*, 60 (6) (1968) 27.
2 N. Ingri, W. Kakolowicz, L. G. Sillén and B. Warnquist, *Talanta*, 14 (1967) 1261.
3 D. Dyrssen, D. Jagner and F. Wengelin, *Computer Calculations of Ionic Equilibria and Titration Procedures*, J. Wiley, New York, 1968.
4 F. Detar, *Computer Programs for Chemistry*, Vol. II, 1969, pp. 65–67.
5 H. P. Meissner, C. L. Kusik and W. H. Datzell, *J. Educ. Chem.* (*Fundamentals*), 8 (1969) 659.
6 A. A. Bugaevski, *J. Anal. Chem. USSR*, 25 (1970) 405.
7 M. Tanaka and G. Nakagawa, *Anal. Chim. Acta*, 33 (1965) 543.