

General Digital Computer Program for Chemical Equilibrium Calculations

Chemical equilibrium problems, especially those involving ionic equilibria, occupy a large portion of the undergraduate chemistry curriculum. The modern approach to the solution of equilibrium problems usually involves writing the pertinent equilibrium constant expressions along with additional equations specifying the conditions of the particular problem, such as material balance equations, the electro-neutrality equation or the proton balance equations, and solving these for the unknown concentrations. Although writing equations specifying the problem is straightforward, the actual solution of these non-linear simultaneous equations often presents difficulties. At the elementary level, these equations are usually solved by making approximations which will make the equations either linear or particularly well-behaved non-linear ones. Unfortunately, as problems become more complex making *a priori* reasonable approximations becomes more difficult. Other approaches to the solution of these equations involve graphical (1) or numerical methods. The numerical methods have received repeated attention [see (2) and the references contained therein], and generally involve algebraic manipulation of the equations to obtain one or more higher degree equations in a few unknowns, which are then solved using well-known numerical methods such as the Newton-Rapheson or iterative procedures (3).

The aim of the present work was to show how a general digital computer program for solving equilibrium problems could be formulated. Use of a systematic trial-and-error numerical method allowed the equilibrium constant and material and charge balance equations to be written in the program in a form closely corresponding to the equations as initially written in the statement of the problem. A program in this form does not require tedious algebraic combinations of equations; this allows a general program in which only the equations to be solved, the values of the constants and analytical concentrations, and the output format must be specified for any given problem. The method used is described below. The program, written in a commonly used computer language, FORTRAN, is given and explained through the use of two examples.

Formulation of the Method

Since a computer is incapable of operating in terms of unknown variables, algebraic equations as such cannot be introduced into the computer. The approach taken then to solve n algebraic equations in n unknowns is to guess values for two variables, which are called, by analogy to graphical methods (1), the *master variables*.

The remaining variables are then calculated using $n-2$ of the equations written in only slightly modified form. This modified form for writing the $n-2$ equations consists of solving each for one of the variables in terms of the others, and sequencing the equations in such a manner that any variable can be computed using the master variables and variables already computed in the sequence. The remaining two equations are used as tests. If both are satisfied when all of the variables are substituted into them, the problem is solved. If they are not satisfied, the master variables are systematically varied until the two test equations are satisfied.

The master variables [called CX and $CY(I)$ in the program] are varied in the following manner. CX is initially given the guessed value and two CY values are found, $CY(1)$ and $CY(2)$, each of which satisfy one of the two test equations. If the correct CX has not been used, $CY(1)$ will not equal $CY(2)$. CX is then varied by a systematic procedure, new values of the variables and $CY(1)$ and $CY(2)$ are calculated, and this procedure is continued until $CY(1)$ equals $CY(2)$. A flow chart of the method is given in Figure 1. Although the

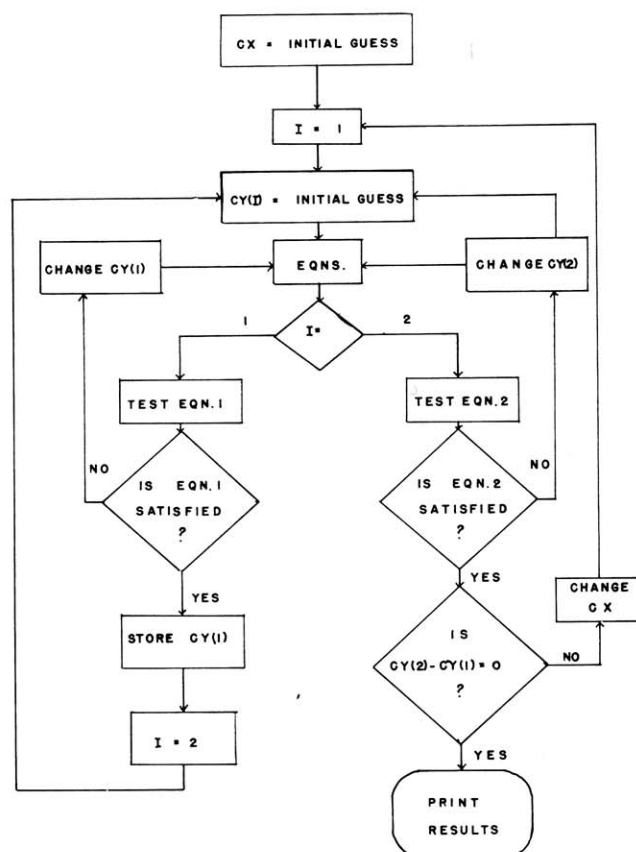


Figure 1. Overall flow chart of computer program for solving equilibrium problems.

systematic trial-and-error procedure used in this program is somewhat slower than iterative or the Newton-Rapheson methods, it is more generally applicable and does not suffer the convergence problems associated with these methods (3). The time required on a high speed digital computer is often negligibly different. The actual procedure of carrying out these operations, the method of making initial guesses for the master variables, and the manner in which the program operates is best illustrated by the solution of a simple problem.

Example 1.

Calculate the concentrations of all species in a *CF* solution a weak acid, H_3A .

(1) *Formulation of Equations.* The six species in this system are H^+ , OH^- , H_3A , H_2A^- , HA^{2-} , and A^{3-} . They are related through the following equations:

$$K_1 = \frac{[H_2A^-][H^+]}{[H_3A]} \quad K_2 = \frac{[HA^{2-}][H^+]}{[H_2A^-]} \quad K_3 = \frac{[A^{3-}][H^+]}{[HA^{2-}]}$$

$$K_w = [H^+][OH^-]$$

$$C = [H_3A] + [H_2A^-] + [HA^{2-}] + [A^{3-}]$$

$$[H^+] = [OH^-] + [H_2A^-] + 2[HA^{2-}] + 3[A^{3-}]$$

(2) *Choosing Master Variables.* Two species, called *CX* and *CY(I)*, are chosen, and initial guesses are made for their concentrations. The basis for the initial guesses will be discussed later, but guesses generally are not critical and are values known to be smaller than the actual amount. In this problem the following guesses were used:

$$CX = [H^+] = 10^{-5} \text{ (initial guess)}$$

$$CY(I) = [A^{3-}] = 10^{-30} \text{ (initial guess)}$$

(3) *Computing Other Variables.* All but two of the equations are used, in a serial fashion, to calculate concentrations of remaining species. Simple algebraic combination of equations may sometimes be necessary.

$$[H_3A] = \frac{[H^+]^3[A^{3-}]}{K_1K_2K_3}$$

$$[HA^{2-}] = \frac{[H^+][A^{3-}]}{K_3}$$

$$[H_2A^-] = \frac{K_1[H_3A]}{[H^+]}$$

$$[OH^-] = \frac{K_w}{[H^+]}$$

Note that the sequence of calculations is important. Since the computer cannot operate in terms of unknown variables, $[H_2A^-]$ cannot be calculated in the third equation unless a value for $[H_3A]$ has been calculated and is stored in the computer.

(4) *Formulation of Test Equations.* The two remaining equations are now written in the following form:

$$Y = [H_3A] + [H_2A^-] + [HA^{2-}] + [A^{3-}] - C$$

and

$$Z = [H_2A^-] + 2[HA^{2-}] + 3[A^{3-}] + [OH^-] - [H^+]$$

and are referred to as test equations 1 and 2.

(5) *Introduction of Problem Into Computer Program.* The complete program for this problem written in FORTRAN is given in Figure 2. While it is beyond the scope of this discussion to delve into the details of programming, a brief explanation of the workings and

statements of the program will be attempted. The essentials of FORTRAN, a widely used algebraic compiler designed for science and engineering problems, can be mastered in a few hours; a good introductory treatment is that of McCracken (4).

The coding of variables and constants is as follows:

$$\begin{array}{ll} [H^+] = CX & K_1 = AK1 \\ [A^{3-}] = CY(I) & K_2 = AK2 \\ [H_3A] = CH3A & K_3 = AK3 \\ [H_2A^-] = CH2A & K_w = WK \\ [HA^{2-}] = CHA & \\ [OH^-] = COH & \end{array}$$

The concentrations of the species are all terms starting with the letter "C". The equilibrium constants use the letter "K" as a second letter but must begin with a letter other than *I, J, K, L, M, or N*, which are reserved for fixed point variables or integers in FORTRAN.

The enclosed portions of the program in Table 1 represent those parts added to the general program concerning the particular problem under consideration.

Part 1. Reading of equilibrium constants and analytical concentrations.

In this case the data given is for a solution in which $C = 0.2F$, $K_1 = 10^{-3}$, $K_2 = 10^{-8}$, and $K_3 = 10^{-13}$. FORTRAN exponential notation is used here, so that 1.0E-3 represents 1.0×10^{-3} .

```

PROGRAM SOLVAR
C = 0.2
Part 1  AK1 = 1.0E-3
        AK2 = 1.0E-8
        AK3 = 1.0E-13
Part 2  WK = 1.0E-14
        CX = 1.0E-5
        LX = 1
        MX = 1
        DIMENSION CY(2)
Part 3  23 I = 1
        15 CY(I) = 1.0E-50
        JY = 1
        LY = 1
        MY = 1
Part 4  11 CH3A = CX**3*CY(I)/(AK1*AK2*AK3)
        CHA = CX*CY(I)/AK3
        CH2A = AK1*CH3A/CX
        COH = WK/CX
        JY = JY + 1
Part 5  GO TO (1,2), I
        1 Y = CH3A + CH2A + CHA + CY(I) - C
        GO TO 7
Part 6  2 Y = CH2A+2.*CHA+3.*CY(I)+COH-CX
        7 IF (JY - 100) 3,3,12
        3 IF (MY - 1) 4,4,5
        4 Z1 = Y
        MY = MY + 1
        5 IF (Z1) 6,12,8
        6 IF (Y) 9,12,10
        8 IF (Y) 10,12,9
        9 CY(I) = CY(I) + 10.0**(-LY)
        GO TO 11
        10 CY(I) = CY(I) - 10.0**(-LY)
        LY = LY + 1
        IF (CY(I)/10.0**(-LY) - 10.0**10) 9,9,12
        12 IF (I-1) 13,13,14
        13 I = I + 1
        GO TO 15
        14 Y = CY(2) - CY(1)
        24 IF (MX - 1) 16,16,17
        16 Z2 = Y
        MX = MX + 1
        17 IF (Z2) 18,19,20
        18 IF (Y) 21,19,22
        20 IF (Y) 22,19,21
        21 CX = CX + 10.0**(-LX)
        GO TO 23
        22 CX = CX - 10.0**(-LX)
        LX = LX + 1
        IF (CX/10.0**(-LX) - 10.0**10) 21,21,19
Part 7  100 FORMAT (7E13.4)
        19 PRINT 100,CX,CY(1),CY(2),CH3A,CH2A,CHA,COH
        END
        END

```

Figure 2. General computer program with statements for Example 1.

Table 1. Intermediate Values of CX , $CY(1)$, $CY(2)$, and Y

CX	$CY(2)$	$CY(1)$	Y (Statement 14)
0.1000E-04	0.9975E-16	0.1978E-11	-0.1978E-11
0.1000E+00	0.9990E-20	0.1978E-21	0.9792E-20
0.1001E-01	0.9981E-19	0.1813E-18	-0.8148E-19
0.2001E-01	0.4995E-19	0.2377E-19	0.2618E-19
0.1101E-01	0.9083E-19	0.1374E-18	-0.4655E-19
0.1201E-01	0.8326E-19	0.1066E-18	-0.2331E-19
0.1301E-01	0.7686E-19	0.8434E-19	-0.7481E-20
0.1401E-01	0.7138E-19	0.6786E-19	0.3518E-20
0.1311E-01	0.7627E-19	0.8247E-19	-0.6196E-20
0.1321E-01	0.7570E-19	0.8065E-19	-0.4954E-20
0.1331E-01	0.7513E-19	0.7884E-19	-0.3709E-20
0.1341E-01	0.7457E-19	0.7718E-19	-0.2609E-20
0.1351E-01	0.7402E-19	0.7552E-19	-0.1499E-20
0.1361E-01	0.7347E-19	0.7390E-19	-0.4283E-21
0.1371E-01	0.7293E-19	0.7233E-19	0.6013E-21
0.1362E-01	0.7342E-19	0.7374E-19	-0.3237E-21
0.1363E-01	0.7337E-19	0.7358E-19	-0.2160E-21
0.1364E-01	0.7331E-19	0.7343E-19	-0.1136E-21
0.1365E-01	0.7326E-19	0.7327E-19	-0.1114E-22
0.1366E-01	0.7321E-19	0.7311E-19	0.9333E-22
0.1365E-01	0.7325E-19	0.7325E-19	0.0000E+00

Part 2. Initial guess for CX .

Part 3. Initial guess for $CY(I)$.

Part 4. Sequence of equations for other variables.

These are the equations in (3) above written in computer language. In FORTRAN the multiplication operation is represented by *, the division operation by / and the exponentiation operation by **.

Part 5. Test equation 1.

Part 6. Test equation 2.

These are the equations in (4) above. Note that the addition and subtraction operations are represented by their conventional symbols.

Part 7. Output instructions.

In this case the instructions are to print the values of the concentrations of all of the species in a specified format in E-notation.

Note that the statement numbers attached to some of the statements in these parts must be included.

The Program

The following is a general description of the manner in which the program is used in the computer to carry out the computation. The program is carried out in steps from top to bottom, but also involves skipping ahead or back depending upon instructions in the program. The program name is read, followed by the input data and initial guesses for CX and $CY(I)$. The parameters LX , MX , JY , LY , and MY , are internal control integers of the program, all initially set to a value of one. The DIMENSION statement indicates that there are two different CY 's used in the program. Values for the concentrations of the various species are now calculated in Part 4, based on the input data and the initial guesses of CX and $CY(I)$.

The statements involving JY are provisions to prevent the computer from increasing CY without limit in case some error has occurred (for example a bad initial guess for CX). The program is designed to increase JY by one every time CY is increased. Statement 7 tests to see if JY is 100. If it is not, the program continues. If it is, the program leaves $CY(I)$ and goes on to calculate $CY(2)$. These steps are included to prevent wasting time on the computer if errors occur or wrong input data is used.

The program now calculates a value of $CY(I)$ which

satisfies test equation one. This is accomplished as follows. A value of Y is calculated in statement 1 based on the initially guessed value of CY and the other concentrations. Since the initial value of CY is guessed smaller than the actual value, the sign of the value of Y just calculated serves to indicate when subsequently calculated CY -values are too small or too large. Therefore this first value of Y is stored as a controlling number $Z1$ in statement 4. The parameter MY and statement 3 assure that $Z1$ is calculated only for the initially guessed CY . Let us assume that Y is negative when CY is too small. Then a negative value of $Z1$ is stored to control the program. Whenever a tested value of CY appears which yields a negative value of Y , the program will cause CY to increase. Whenever a positive value of Y appears, the program will cause CY to decrease. The correct value of CY will be that value for which Y is zero. Y is tested by statements 6 or 8, depending upon the sign of $Z1$. If $CY(I)$ is too small, as it must be on the first trial, it is increased by 0.1 in statement 9 (LY is initially equal to one) and the program cycles back to statement 11 to try this new $CY(I)$. As long as $CY(I)$ is too small it is increased in steps of 0.1. When Y changes sign, indicating $CY(I)$ is now too large, $CY(I)$ is decreased by 0.1 in statement 10, and increased by 0.01 in statement 9 (LY has changed to two following statement 10). This process continues until essentially ten significant figures have been accumulated for $CY(I)$, as tested by the IF-statement two statements after statement 10.

Now $CY(2)$ is treated in an identical manner using test equation 2. The change from $CY(I)$ to $CY(2)$ is accomplished using statements 12 and 13. After a value of $CY(2)$ which satisfies test equation 2 (i.e., minimizes that value of Y) is calculated, the difference between $CY(I)$ and $CY(2)$ is calculated in statement 14. Now CX is varied using statements 24 on by a procedure virtually identical to that used for changing CY . For each new value of CX two values of CY must be calculated using the above procedure. At last, when a value of CX which minimizes the value of Y in statement 14 has been calculated, all of the equations are simultaneously satisfied, and the answers are printed.

To illustrate how the program operated in this problem, values of $CY(I)$, $CY(2)$ and Y (in statement 14) for every CX were printed and these results are shown in Table 1. For the initial CX (10^{-5}) $CY(I)$ and $CY(2)$ were calculated (the individual CY -values used in calculating $CY(I)$ and $CY(2)$ are not given) and Y was negative. CX was increased to 0.1, etc. Note that CX is increased until Y changes sign. Although these values are calculated to ten significant figures, only the first four are shown. The computer program and technique for varying CX and CY is similar to that used by Herman and Bard for solving one-variable equations (5). Although this method appears to be time-consuming, the total compiling and computation time for this problem on a Control Data Corporation 1604 computer was only 17 seconds and the following results were obtained:

$$\begin{aligned}
 [H^+] &= 1.365 \times 10^{-2}M & [H_2A^-] &= 1.365 \times 10^{-2}M \\
 [A^{3-}] &= 7.325 \times 10^{-22}M & [HA^{2-}] &= 1.0 \times 10^{-8}M \\
 [H_3A] &= 0.1863M & [OH^-] &= 7.325 \times 10^{-13}M
 \end{aligned}$$

Discussion

The variables selected to be represented as CX and $CY(I)$ are generally those which appear most often in the equations. Thus, one generally selects $[H^+]$ and complexing ligands since these repeatedly appear in equilibrium expressions. To solve directly for the remaining variables it may be necessary to combine some equations, although this is of minor difficulty.

The selection of the initial CX and $CY(I)$ can be illustrated with Figure 3. $CY(2)$ and $CY(1)$ are plotted against CX . The intersection represents the correct values of $CY(I)$ and CX . To the left of this point Y is negative, to the right it is positive. The program is written so that scanning is done in the increasing direction. The initial CX therefore must be smaller than the correct value and the initial $CY(I)$ must fall below both lines (such as point one). An initial CX close to the point of intersection allows the initial $CY(I)$ to be guessed with a greater probability that it will fall below both lines. For example if CX is guessed too small, it is possible that the initial CY chosen will be above one of the CY lines at that point.

Another difficulty arises in the inability of the computer to retain more than eight or nine significant figures. In writing the sequence of equations one must be certain that a very low concentration isn't calculated as the difference between two large numbers. Thus, it would be poor to use the following equation:

$$C = [H_3A] + [H_2A^-] + [HA^{2-}] + [A^{3-}]$$

to calculate $[A^{3-}]$. This equation should be used either for calculating larger concentrations or as a test equation (as was done here).

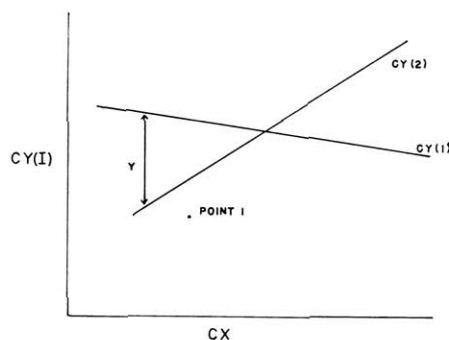


Figure 3. Plot of $CY(I)$ versus CX .

Example 2

Consider a solution which is 0.01 M in Ni^{2+} , 0.1 M in NH_3 , 0.1 M in NH_4^+ , and C M in Na_2H_2Y , where H_4Y is ethylenediamine-tetraacetic acid (EDTA). This problem involves 17 solution species, including six nickel-ammonia complexes, the nickel-EDTA complex, and five ionization states of H_4Y . The various equilibrium constants are given below (5).

Equilibrium Constants (Part 1):

Formation constants for $Ni(NH_3)_6^{2+}$:

$$K_1 = [Ni(NH_3)_2^{2+}]/[Ni^{2+}][NH_3] = 562$$

$$K_2 = 159.0$$

$$K_3 = 48.0$$

$$K_4 = 14.1$$

$$K_5 = 5.12$$

$$K_6 = 0.98$$

Ionization Constants for H_4Y :

$$K_{a1} = 1.0 \times 10^{-2}$$

$$K_{a2} = 2.14 \times 10^{-3}$$

$$K_{a3} = 6.97 \times 10^{-7}$$

$$K_{a4} = 5.50 \times 10^{-11}$$

$$K_{NiY} = 4.17 \times 10^{18}$$

$$K_b \text{ (for } NH_3) = 1.8 \times 10^{-5}$$

$$K_w = 10^{-14}$$

Variables and Initial Values:

$$\text{(Part 2) } CX = [NH_3] = 0.05$$

$$\text{(Part 3) } CY(I) = [Ni^{2+}] = 1.0 \times 10^{-50}$$

Sequence of Equations (Part 4):

$$[Ni(NH_3)^{2+}] = K_1[Ni^{2+}][NH_3]$$

$$[Ni(NH_3)_2^{2+}] = K_2[Ni(NH_3)^{2+}][NH_3]$$

$$[Ni(NH_3)_3^{2+}] = K_3[Ni(NH_3)_2^{2+}][NH_3]$$

$$[Ni(NH_3)_4^{2+}] = K_4[Ni(NH_3)_3^{2+}][NH_3]$$

$$[Ni(NH_3)_5^{2+}] = K_5[Ni(NH_3)_4^{2+}][NH_3]$$

$$[Ni(NH_3)_6^{2+}] = K_6[Ni(NH_3)_5^{2+}][NH_3]$$

$$[NiY^{2-}] = 0.01 - [Ni^{2+}] - [Ni(NH_3)^{2+}] - [Ni(NH_3)_2^{2+}] - [Ni(NH_3)_3^{2+}] - [Ni(NH_3)_4^{2+}] - [Ni(NH_3)_5^{2+}] - [Ni(NH_3)_6^{2+}]$$

$$[NH_4^+] = 0.2 - [NH_3] - [Ni(NH_3)^{2+}] - 2[Ni(NH_3)_2^{2+}] - 3[Ni(NH_3)_3^{2+}] - 4[Ni(NH_3)_4^{2+}] - 5[Ni(NH_3)_5^{2+}] - 6[Ni(NH_3)_6^{2+}]$$

$$[OH^-] = K_b \frac{[NH_3]}{[NH_4^+]}$$

$$[H^+] = \frac{K_w}{[OH^-]}$$

$$[Y^{4-}] = \frac{[NiY^{2-}]}{K_{NiY}[Ni^{2+}]}$$

$$[HY^{3-}] = \frac{[H^+][Y^{4-}]}{K_{a4}}$$

$$[H_2Y^{2-}] = \frac{[H^+][HY^{3-}]}{K_{a3}}$$

$$[H_3Y^-] = \frac{[H^+][H_2Y^{2-}]}{K_{a2}}$$

$$[H_4Y] = \frac{[H^+][H_3Y^-]}{K_{a1}}$$

Test Equations:

$$\text{(Part 5) } 1. Y = C - [H_4Y] - [H_3Y^-] - [H_2Y^{2-}] - [HY^{3-}] - [NiY^{2-}] - [Y^{4-}]$$

$$\text{(Part 6) } 2. Y = 2[Ni^{2+}] + 2[Ni(NH_3)^{2+}] + 2[Ni(NH_3)_2^{2+}] + 2[Ni(NH_3)_3^{2+}] + 2[Ni(NH_3)_4^{2+}] + 2[Ni(NH_3)_5^{2+}] + 2[Ni(NH_3)_6^{2+}] + [NH_4^+] + [H^+] + 2C - [H_3Y^-] - 2[H_2Y^{2-}] - 3[HY^{3-}] - 4[Y^{4-}] - [OH^-] - 0.12 - 2[NiY^{2-}]$$

Table 2 shows the calculated results for four values of C . The program time was 80 seconds. Note that this

Table 2. Results for Example 2^a

	$C = 0$	$C = 0.005$	$C = 0.01$	$C = 0.012$
$[Ni^{2+}]$	3.17×10^{-6}	1.11×10^{-6}	2.44×10^{-12}	2.02×10^{-17}
$[Ni(NH_3)^{2+}]$	1.16×10^{-4}	4.51×10^{-5}	1.10×10^{-10}	8.85×10^{-16}
$[Ni(NH_3)_2^{2+}]$	1.21×10^{-3}	5.18×10^{-4}	1.40×10^{-9}	1.10×10^{-14}
$[Ni(NH_3)_3^{2+}]$	3.86×10^{-3}	1.83×10^{-3}	5.47×10^{-9}	4.18×10^{-14}
$[Ni(NH_3)_4^{2+}]$	3.55×10^{-3}	1.86×10^{-3}	6.17×10^{-9}	4.59×10^{-14}
$[Ni(NH_3)_5^{2+}]$	1.19×10^{-3}	6.90×10^{-4}	2.53×10^{-9}	1.83×10^{-14}
$[Ni(NH_3)_6^{2+}]$	7.60×10^{-5}	4.88×10^{-5}	1.98×10^{-10}	1.40×10^{-15}
$[NH_4^+]$	0.100	0.110	0.120	0.122
$[NH_3]$	0.0653	0.0722	0.0800	0.0779
$[H_4Y]$	0	6.80×10^{-31}	5.82×10^{-25}	8.39×10^{-20}
$[H_3Y^-]$	0	8.03×10^{-24}	6.98×10^{-18}	9.63×10^{-12}
$[H_2Y^{2-}]$	0	2.03×10^{-16}	1.79×10^{-11}	2.37×10^{-6}
$[HY^{3-}]$	0	1.66×10^{-14}	1.49×10^{-8}	1.88×10^{-3}
$[Y^{4-}]$	0	1.08×10^{-14}	9.82×10^{-10}	1.19×10^{-4}
$[H^+]$	8.51×10^{-10}	8.46×10^{-10}	8.34×10^{-10}	8.71×10^{-10}
$[OH^-]$	1.17×10^{-5}	1.18×10^{-5}	1.20×10^{-5}	1.15×10^{-5}
$[NiY^{2-}]$	0	0.005	0.01	0.01

^a Time 80 seconds.

program can be used conveniently to calculate concentrations for many different conditions by using DO-loops for generating data (e.g., C in the case above). For example the calculation in Table 2 could represent four points in the titration of nickel(II) with EDTA in an ammoniacal medium. Similarly, to calculate the concentrations of the various species during the titration of H_3A with sodium hydroxide only the second test equation need be changed to

$$Y = [H_2A^-] + 2[HA^{2-}] + 3[A^{3-}] + [OH^-] - [H^+] - B$$

where B represents the moles of sodium hydroxide added per liter. A DO-loop for generating various values of B would then calculate the titration curve of H_3A with sodium hydroxide. Library programs are available (7) to allow automatic plotting of the titration curve by the computer.

The program written here, or modifications of it, should prove useful for solving a large number of different kinds of problems in chemical equilibrium and for certain types of simultaneous, non-linear algebraic equations. The program presented must not be used without some background in computer programming.

For example, there are some variations in versions of FORTRAN for different computers which would require modification of the program presented. Appendix 1 in McCracken (4) might be useful for rewriting this program in another version of FORTRAN.

Literature Cited

- (1) SILLEN, L. A., in "Treatise on Analytical Chemistry," KOLTHOFF, I. M., AND ELVING, P. J., editors, Part I, Vol. I, Interscience, New York, 1959. Ch. 8.
- (2) (a) ANTHONY, R. G., AND HIMMELBLAU, D. M., *J. Phys. Chem.*, **67**, 1080 (1963).
(b) SWINNERTON, J. W., AND MILLER, W. W., *J. CHEM. EDUC.*, **36**, 485 (1959).
- (3) SCARBOROUGH, J. B., "Numerical Mathematical Analysis," The Johns Hopkins Press, Baltimore, 1962, p. 213.
- (4) MCCRACKEN, D. D., "A Guide to FORTRAN Programming," John Wiley & Sons, New York, 1961.
- (5) HERMAN, H. B., AND BARD, A. J., *Anal. Chem.*, **35**, 1121 (1963).
- (6) LAITINEN, H. A., "Chemical Analysis," McGraw-Hill, N. Y., 1960, pp. 228-230.
- (7) For example, UTM2-02-033, "Point plot tape generator for CDC 1604 digital computer to the CDC 165 digital plotter," The University of Texas Computation Center, Austin, Texas (1962).