Transactions of the Illinois Academy of Science
(1984), Volume 77, 1 and 2, pp. 61-66

A MICROCOMPUTER PROGRAM FOR PERFORMING JOB'S ANALYSIS OF A COMPLEX

J.E. House, Jr., and D.K. Tcheng
Department of Chemistry
Illinois State University
Normal, Illinois 61761

ABSTRACT

A microcomputer program is described which determines the composition of a complex by Job's method and, using a graphics plotter, draws the Job's plot.

INTRODUCTION

In a number of types of situations, it is necessary to determine a break point in a curve that represents some property of a system as a function of concentration. One such situation is that of applying Job's method to the problem of determining the composition of a metal complex (Angelici, 1977; Jones, 1964; Vosburgh and Cooper, 1941). If the equilibrium can be represented as

$$M + nL \rightleftarrows ML_n \qquad (1)$$

where n = 1, 2, 3, . . ., the problem is to determine the number (n) of ligands (L) per metal ion (M). Actually, any electron donor-acceptor interaction that leads to complex formation can be considered as well (Gur'yonova, et al., 1975). In this method, some property of the complex, usually absorbance at some appropriate wavelength, is measured as the concentration of the metal, CM, and the concentration of the ligand, CL, are varied while keeping the total concentration of solution, C, fixed. It can readily be shown that the maximum concentration of the complex, and hence the maximum in the measured property, occurs when the ratio of the concentration of metal to that of ligand in the solution is exactly the same as the ratio present in the complex (Angelici, 1977; Jones, 1964). Thus, in principle, varying the ratio CM/CL in a series of isomolar solutions while measuring the appropriate property of the complex should produce a curve having a maximum (or in some cases a minimum). From the position of this maximum, CM and CL of the solution containing the highest concentration of complex is identified and n is determined. Typically, the curves are sketched from experimental data and the maximum is obtained from the graph. We present here a program in BASIC for

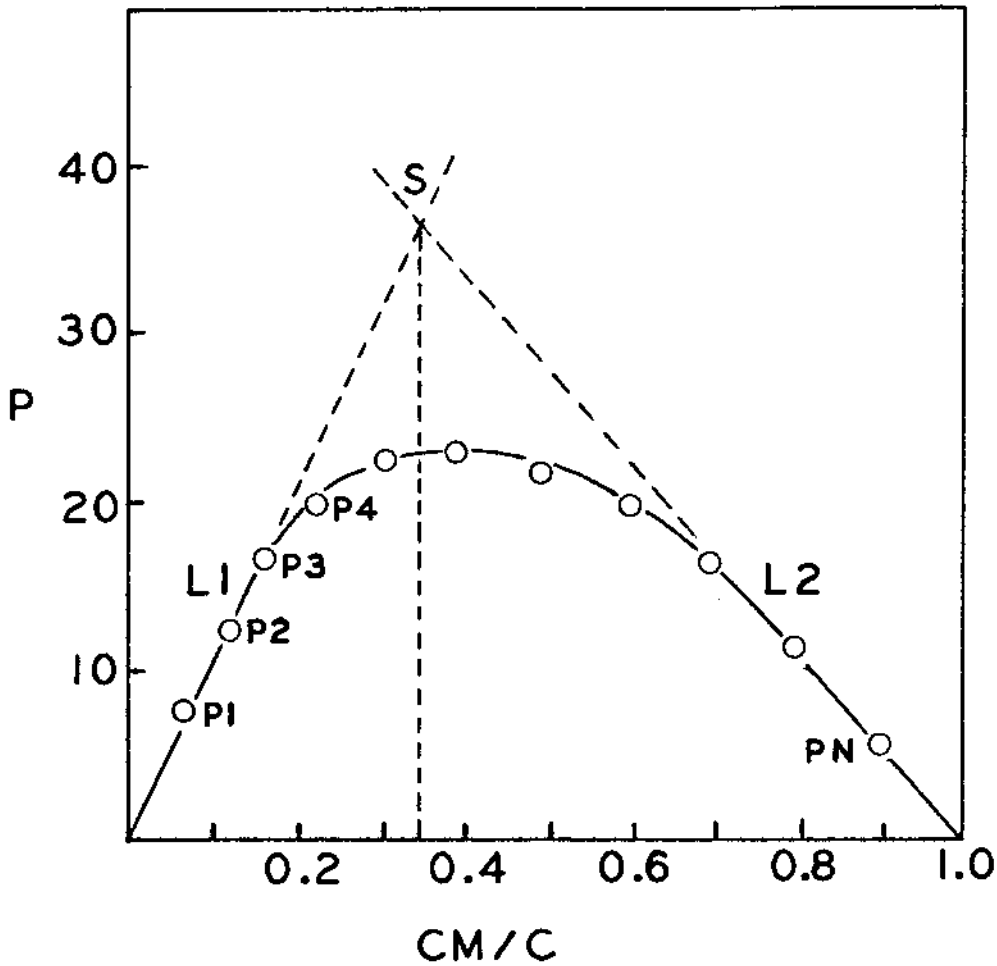

Fig. 1. A typical Job's plot.

simultaneously drawing the curves and locating the maximum using the intersection point of lines established using linear regression.

PRINCIPLES AND ALGORITHM

In practice, a curve from a Job's method plot may have a general shape like that shown in Figure 1. To determine the maximum, the first few data points for the lower CM_i values to the left of the maximum are considered to form one linear portion (L1) while those to the right are considered to form another (L2). The maximum is established by extending these two linear regions until they intersect at point S. A program for carrying out the procedure was written in BASIC for implementation on the Radio Shack PC-2 pocket computer and its graphics plotter. The program is divided into two sections. The first section includes a subroutine to perform linear regression and calculates the regression lines L1 and L2 as well as their point of intersection. The second section graphically illustrates the data used and draws the regression lines and the property vs. concentration curve.

In determining the regression lines, the computation is first carried out with the two data points for the lowest CM values (P1 and P2 of Figure 1) and establishes a linear fit having a correlation coefficient r for those two points. For only two data points, /r/ = 1 so the next higher CM data point (P3) is included. Linear regression is again performed and the value of /r/ is obtained. Now a sufficiency test for linearity is used in the form of a minimum correlation coefficient, D. If /r/ D, the next point is included and /r/ is again determined and compared to D. When adding one more point causes /r/ $<$D, the line L1 is not sufficiently close to linear, and the previous relationship for L1 is used. Thus, each slope and intercept must be saved until the subsequent cycle is complete. The linear relationship for L2 is established in a similar manner including as many data points as possible as long as /r/ $>$D. Thus, in the region L1,

$$P = m_1(CM) + b_1 \tag{2}$$

and in the region L2,

$$P = m_2(CM) + b_2 \tag{3}$$

where m_1 and m_2 are the slopes and b_1 and b_2 are the intercepts of the two lines. At the point of intersection,

$$CM = (b_2 - b_1)/(m_1 - m_2) \tag{4}$$

Now the ratio of metal to ligand in the complex, M/L, is given by CM/(C − CM) and the ratio of ligand to metal, L/M is given by (C − CM)/CM.

Having determined the composition of the complex, the program now executes the plotting procedure. First, coordinate axes are drawn and labeled, and the numerical values for D, S, and L/M are printed. With the frame in place, the data points are plotted as asterisks and they are connected with a solid line. Next, the dotted lines are drawn that show the linear relationships established for L1 and L2 using dat for which /r/ $>$D. Then, a vertical dotted line is drawn from the intersection point S to the concentration axis. A complete program listing and a data dictionary are given in the Appendix.

COMPUTATION

In order to show the input/output characteristics of the program, we present here an example computation. Table 1 shows the data used in this computation. The data are given in the program in lines 120-220.

Table 1. Data for program testing. Total concentration is 1.00 *M*.

CM, Moles/liter	P	CM, Moles/liter	P
0.065	7.4	0.500	21.0
0.110	12.5	0.600	18.5
0.160	18.0	0.695	17.0
0.210	19.4	0.790	11.3
0.300	22.0	0.900	5.2
0.402	21.9		

In the present case, the value of D = 1.00 was used first, assuming the L1 and L2 must have exactly a linear fit. Figure 2(a) shows the result obtained. Next, a value

of D = 0.99 was used with the result shown in Figure 2(b). Successively smaller values of D were used until a value of 0.90 was reached. While in this particular case the ratio L/M could be identified as 2, it is obvious (from Figure 2(f)) that a correlation coefficient of 0.90 does not provide a very good fit to the data. Unless the data are subject to large experimental errors, it appears that a D value of at least 0.98 should be chosen. It is, therefore, preferable to have a large number of data points and to have as accurate data as possible.

In using the program with other data, the value of D must be entered at line 100 and the number of data points must be entered at line 110. The data points used must be entered at lines 120-220.

Each of the diagrams shown in Figure 2(a)-(f) requires about 40 sec for the computation and drawing of the figure. Therefore, it is possible to apply Job's method in a much more convenient and rapid manner than has been available previously.

LITERATURE CITED

Angelici, R.L. 1977. Synthesis and Technique in Inorganic Chemistry. Saunders, Philadelphia. xiv + 237 pp.

Gur'yonova, E.N., I.P. Gol'dshtein, and I.P. Romm. 1975. Donor-Acceptor Bond. Translated by R. Kondor. Wiley, New York. ix + 366 pp.

Jones, M.M. 1964. Elementary Coordination Chemistry. Prentice-Hall, Englewood Cliffs. xiii + 473 pp.

Vosburgh, W.C., and G.R. Cooper. 1941. Complex ions. I. The identification of complex ions in solution by spectrophotometric measurements. J. Amer. Chem. Soc. 63:443.

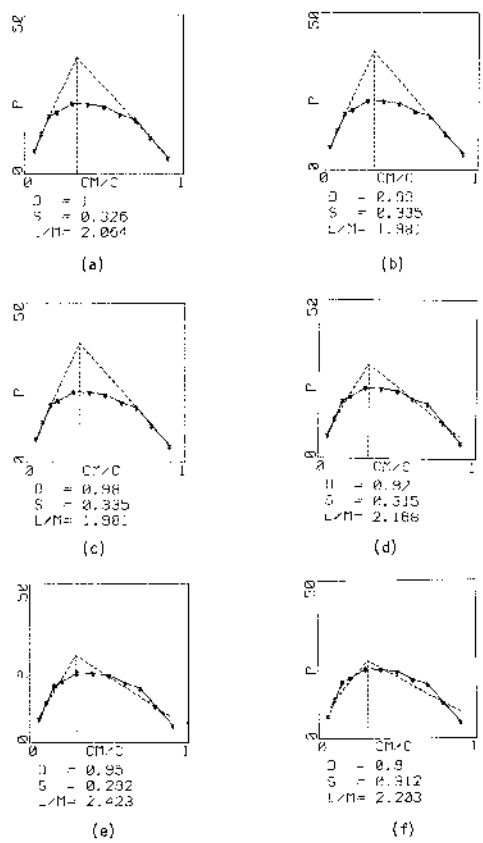

Fig. 2. Output obtained using data shown in Table 1.

APPENDIX

PROGRAM DATA DICTIONARY

Name	Description
BP	X coordinate of break point
CC	correlation coefficient
D1	domain lower bound
D2	domain upper bound
I	data element index
II	index increment
K	constant, value = 200
L	loop counter
LI	Y intercept for left correlation line
LS	slope for left correlation line
LI	loop counter 1
NE	number of elements used for corr. line
NP	number of data points
P	array of data points DIM (NP,2)
RI	Y intercept for right correlation line
RS	slope for right correlation line
R1	range lower bound
R2	range upper bound
S	slope of correlation line from subroutine
S1	sum of XY
S2	sum of X**2
S3	sum of Y**2
S4	sum of X
S5	sum of Y
X	X coordinate to be plotted
X1	last X coordinate plotted
Y	Y coordinate to be plotted
YI	Y intercept calculated in subroutine
Y1	last Y coordinate plotted

PROGRAM LISTING

```
100:DATA .99
110:DATA 11
120:DATA .065,7.4
130:DATA .110,12.5
140:DATA .160,18.0
150:DATA .210,19.4
160:DATA .300,22.0
170:DATA .402,21.9
180:DATA .500,21.0
190:DATA .600,18.5
200:DATA .695,17.0
210:DATA .790,11.3
220:DATA .900,5.2
230:READ D
240:READ NP
250:DIM P(NP,2)
260:FOR L=1TO NP
270:READ P(L,1)
280:READ P(L,2)
290:NEXT L
300:I=1
310:II=1
320:GOSUB 430
330:RS=S
340:RI=YI
350:I=NP
360:II=-1
370:GOSUB 430
380:LS=S
390:LI=YI
400:BP=(LI-RI)/(RS
    -LS)
410:PRINT "BP=";BP
420:GOTO 630
430:REM
     CORRELATION
        LINE
     SUBROUTINE
440:S1=0
450:S2=0
460:S3=0
470:S4=0
480:S5=0
490:NE=0
500:NE=NE+1
510:S1=S1+P(I,1)*P
    (I,2)
520:S2=S2+P(I,1)*P
    (I,1)
530:S3=S3+P(I,2)*P
    (I,2)
540:S4=S4+P(I,1)
550:S5=S5+P(I,2)
560:I=I+II
570:IF NE<2THEN
    GOTO 500
580:CC=ABS (NE*S1-
    S4*S5)/SQR((NE*S
    2-S4*S4)*(NE*S
    3-S5*S5))
590:IF CC<DTHEN
    RETURN
600:S=(NE*S1-S4*S5
    )/(NE*S2-S4*S4
    )
610:YI=(S5-S*S4)/N
    E
620:GOTO 500
630:REM
     GRAPHIC PART
      OF PROGRAM
640:COLOR 0
650:CSIZE 2
660:GRAPH
670:GLCURSOR (16,-
    250)
680:SORGN
690:K=200
700:D1=0
710:D2=1
720:R1=0
730:R2=50
740:ROTATE 3:
    GLCURSOR (-3,0
    ):LPRINT "0
       P        50"
750:ROTATE 0:
    GLCURSOR (0,-1
    6):LPRINT "0
       CM/C       1
    "
760:LINE (0,0)-(K,
    K),0,,B
770:GLCURSOR (0,-4
    0):LPRINT " D
     =";D
780:GLCURSOR (0,-6
    0):LPRINT " S
     =";INT (BP*10
    00)/1000
790:GLCURSOR (0,-8
    0):LPRINT " L/
    M=";INT ((1-BP
    )/BP*1000)/100
    0
800:Y1=99999999
810:FOR L1=1TO NP
820:X=P(L1,1)
830:Y=P(L1,2)
840:IF Y>R2OR Y<R1
    OR Y1>R2OR Y1<
    R1THEN GOTO 87
    0
850:GLCURSOR ((X1-
    D1)/(D2-D1)*K,
    (Y1-R1)/(R2-R1
    )*K)
860:LINE -((X-D1)/
    (D2-D1)*K,(Y-R
    1)/(R2-R1)*K),
    0
870:CSIZE 1
880:GLCURSOR ((X-D
    1)/(D2-D1)*K-3
    ,(Y-R1)/(R2-R1
    )*K-4)
890:LPRINT "*"
900:X1=X
910:Y1=Y
920:NEXT L1
930:CSIZE 2
940:GLCURSOR ((P(1
    ,1)-D1)/(D2-D1
    )*K,(P(1,1)*RS
    +RI-R1)/(R2-R1
    )*K)
950:LINE -((BP-D1)
    /(D2-D1)*K,(BP
    *RS+RI-R1)/(R2
    -R1)*K),2
960:LINE -((P(NP,1
    )-D1)/(D2-D1)*
    K,(P(NP,1)*LS+
    LI-R1)/(R2-R1)
    *K)
970:GLCURSOR ((BP-
    D1)/(D2-D1)*K,
    (BP*RS+RI-R1)/
    (R2-R1)*K)
980:LINE -((BP-D1)
    /(D2-D1)*K,(0-
    R1)/(R2-R1)*K)
    ,2
```