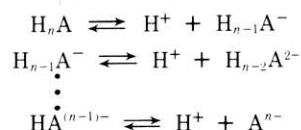


# A General Acid-Base Titration Curve Computer Program

A Fortran program has been developed that will plot on a line printer the titration curve for the titration of any acid or base (weak or strong) with up to eight dissociation constants. The exact equation describing the system is solved to the degree of accuracy specified by the user. The volumes and concentrations of the solutions involved can also be specified.

For an acid with  $n$  dissociable hydrogen ions the following equilibria exist.



The following expressions for equilibrium constants hold. It is assumed that activities of the various species are equal to molar concentrations

$$\begin{aligned} K_1 &= \frac{[\text{H}^+][\text{H}_{n-1}\text{A}^-]}{[\text{H}_n\text{A}]} \\ K_2 &= \frac{[\text{H}^+][\text{H}_{n-2}\text{A}^{2-}]}{[\text{H}_{n-1}\text{A}^-]} \\ &\vdots \\ K_n &= \frac{[\text{H}^+][\text{A}^{n-}]}{[\text{HA}^{(n-1)-}]} \end{aligned}$$

The equilibrium for the self-dissociation of water also holds.

$$K_w = [\text{H}^+][\text{OH}^-] \quad (1)$$

Assuming a mono-acidic base is being used for the titration (e.g., NaOH), the following expression holds for the charge balance of all the species during titration.

$$[\text{Na}^+] + [\text{H}^+] - [\text{OH}^-] = [\text{H}_{n-1}\text{A}^-] + 2[\text{H}_{n-2}\text{A}^{2-}] + \dots + n[\text{A}^{n-}] \quad (2)$$

If the total concentration of all acid species is  $C_a$  then the following mass balance expression holds

$$C_a = [\text{H}_n\text{A}] + [\text{H}_{n-1}\text{A}^-] + \dots + [\text{A}^{n-}] \quad (3)$$

These are the usual equations describing this kind of titration. The next step is to convert the equations into a function that can be efficiently solved by the computer for  $[\text{H}^+]$  for any given value of  $n$ ,  $K_1, \dots, K_n$ ,  $C_a$ ,  $[\text{Na}^+]$ , and  $K_w$ . From the equilibrium constants expressions one gets

$$\begin{aligned} [\text{H}_{n-1}\text{A}^-] &= \frac{K_1[\text{H}_n\text{A}]}{[\text{H}^+]} \\ [\text{H}_{n-2}\text{A}^{2-}] &= \frac{K_2[\text{H}_{n-1}\text{A}^-]}{[\text{H}^+]} = \frac{K_1K_2[\text{H}_n\text{A}]}{[\text{H}^+]^2} \\ &\vdots \\ [\text{A}^{n-}] &= \frac{K_1K_2 \dots K_n[\text{H}_n\text{A}]}{[\text{H}^+]^n} \end{aligned}$$

Now by dividing eqn. (2) by eqn. (3) and substituting in the above relationships one can cancel  $[\text{H}_n\text{A}]$  from the numerator and denominator on the right side to get

$$\frac{[\text{Na}^+] + [\text{H}^+] - [\text{OH}^-]}{C_a} =$$

$$\frac{\frac{K_1}{[\text{H}^+]} + \frac{2K_1K_2}{[\text{H}^+]^2} + \dots + \frac{nK_1K_2 \dots K_n}{[\text{H}^+]^n}}{1 + \frac{K_1}{[\text{H}^+]} + \dots + \frac{K_1 \dots K_n}{[\text{H}^+]^n}}$$

This can be rearranged with eqn. (1) substituted in to give

$$0 = \left[ \frac{\sum_{i=1}^n \left( i[\text{H}^+]^{n-i} \prod_{k=1}^i K_k \right)}{[\text{H}^+]^n + \sum_{i=1}^n \left( [\text{H}^+]^{n-i} \prod_{k=1}^i K_k \right)} \times C_a \right] - [\text{Na}^+] - [\text{H}^+] + \frac{K_w}{[\text{H}^+]} \quad (4)$$

This function looked like it would be straightforward to use, as the sums and products could be easily programmed in terms of  $n$ .

The method used for finding the root corresponding to  $[\text{H}^+]$  was the Newton-Raphson method. An initial guess for  $[\text{H}^+]$  is made and a new value of  $[\text{H}^+]$  is calculated using

$$[\text{H}^+]_{j+1} = [\text{H}^+]_j \times \frac{f([\text{H}^+]_j)}{f'([\text{H}^+]_j)}$$

where  $f$  and  $f'$  are function (4) and its first derivative. A test is used to determine the accuracy of the new value and the calculation is repeated using each new value until the accuracy is as desired. This test and the initial guess are discussed later. This method failed to find the root of function (4). Function (4) was then converted into a polynomial of degree  $n + 2$  whose roots are identical to function (4) by multiplying both sides of (4) by the product of the two denominators.

$$0 = [\text{H}^+]^{n+2} + ([\text{Na}^+] + K_1)[\text{H}^+]^{n+1} + ([\text{Na}^+]K_1 - C_aK_1 + K_1K_2 - K_w)[\text{H}^+]^n + \sum_{i=1}^n \left\{ ([\text{Na}^+] \prod_{k=1}^{i+1} K_k - (i+1)C_a \prod_{k=1}^{i+1} K_k + \prod_{k=1}^{i+2} K_k - K_w \prod_{k=1}^i K_k) [\text{H}^+]^{n-i} \right\}$$

This form is easy to use in the computer but is correct only if  $K_{n+1}$  and  $K_{n+2}$  are equal to zero. The actual dissociation constants go only from  $K_1$  to  $K_n$ . The derivative of this function is especially easy to evaluate. The polynomial behaved very well and roots were readily found using the Newton-Raphson method.

A polynomial of the same form can be derived for the titration of a base. The root however corresponds to  $[\text{OH}^-]$ . This can be easily converted to  $\text{pH}$ . Thus to titrate either an acid or a base, one only needs to decide at the end of the equation solving process if  $[\text{H}^+]$  or  $[\text{OH}^-]$  need be converted to  $\text{pH}$ .

In principle there is no limit to the number of dissociation constants allowed in the calculation. There are, however, two practical limits. Terms like  $K_1K_2 \dots K_n$  in the polynomial tend to be very small for weak acids. Because the computer itself limits how small a number can be stored, a limit is imposed on how many  $K$ 's can be used. If, for example, the smallest number that can be stored in

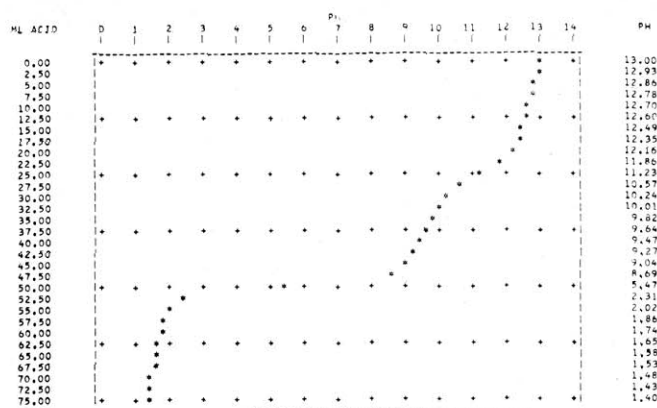


Figure 1. Titration of 50 ml of 0.1 M zinc hydroxide (two dissociation constants) with 75 ml of 0.2 M acid in aliquots of 2.5 ml.

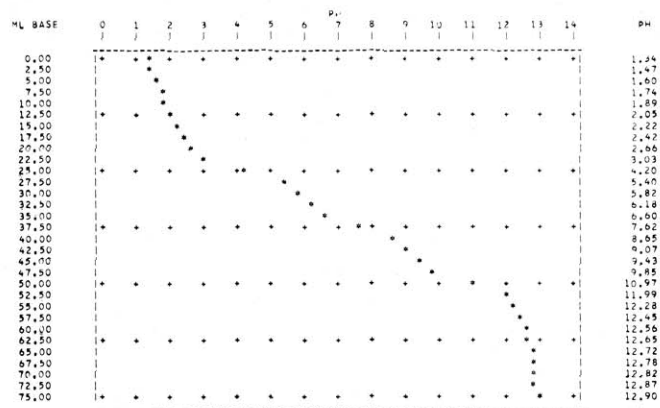


Figure 2. Titration of 50 ml of 0.1 M pyrophosphoric acid (four dissociation constants) with 75 ml of 0.4 M base in aliquots of 2.5 ml.

the computer is  $10^{-75}$  and the average  $K$  equals  $10^{-5}$ , then the computer will limit the calculations to about 15 constants. Another factor limits the number of constants used still further. As the number of constants goes up so does the degree of the polynomial describing the system. A 10th degree polynomial seems to be the highest this method solves consistently for the complete titration. Thus, one is limited to about eight constants. For some values of the input variables, higher degree polynomials have been solved but one never knows ahead of time if the program will be successful in these cases. Both of these limitations could possibly be overcome by using a different function than the  $n + 2$  degree polynomial but it is not obvious how to develop one.

The input to the program consists of the following

Heading (reproduced at the beginning of each plot)  
 Number of dissociation constants  
 Indicator telling if acid or base is being titrated  
 Indicator telling if  $K$ 's or  $pK$ 's are being input  
 Molar concentration of material being titrated  
 Molar concentration of titrant  
 Volume in ml of material being titrated  
 Total volume in ml of titrant to be added  
 Aliquot size of titrant in ml  
 Accuracy of equation solution  
 Maximum number of iterations allowed in equation solution (in case of divergence)  
 Dissociation constants or  $pK$ 's

All input, except the heading, number of constants, and the constants themselves can be left blank and the program will supply preset values.  $K_w$  is set equal to  $10^{-14}$  by the program.

Strong acids and bases are handled by putting in a large equilibrium constant (e.g.,  $10^4$ ). The initial guess for  $[H^+]$  at the first point of the titration is made using the usual approximation for weak acids and bases:  $[H^+] = \sqrt{K C_a}$ . This is used even for strong acids or bases. Even though this guess is very poor in this case, the correct solution to the equation is still found although a larger number of iterations is needed. If several dissociation constants are input, these are automatically sorted and the largest is used in the approximation. The initial guess for all other points in the titration is just  $[H^+]$  for the previous point. This is a poor guess if  $pH$  is changing rapidly but the solution to the equation is still found by using more iterations than normally needed.

The test for convergence in the solution of the equation compares the new value of  $[H^+]$  with the previous value by dividing the difference by the average of the two values.

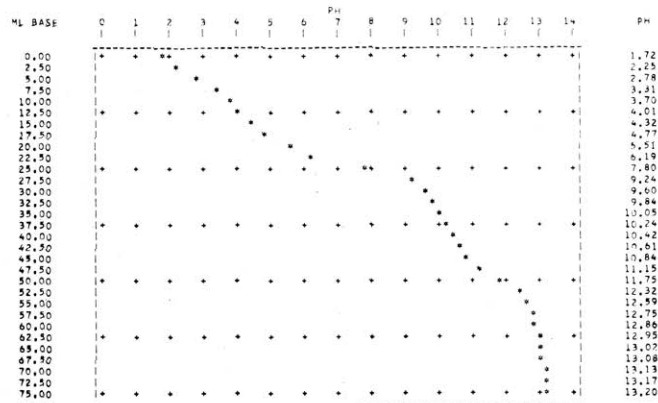


Figure 3. Titration of 50 ml of 0.1 M lysyl aspartyl histidinyl cysteinyl tyrosyl glutamic acid (eight dissociation constants) with 75 ml of 0.8 M base in aliquots of 2.5 ml.

$$E = \frac{|[H^+]_{j+1} - [H^+]_j|}{([H^+]_{j+1} + [H^+]_j)/2}$$

When  $E$  becomes less than the accuracy indicated in the input data, the iterations stop and the point is plotted. Normally about 2-6 iterations are needed to get  $E$  below  $10^{-4}$ .

Output consists of the heading and all data input or preset by the program. The plot itself is restrained between  $pH = 0$  and  $pH = 14$ . After the addition of each aliquot of titrant, the total titrant added is output and the  $pH$  is plotted. Also, output at each point is the value of the  $pH$  and the number of iterations needed to solve the equation at that point. If the  $pH$  goes outside the 0-14 scale, "<" or ">" is printed at the appropriate edge so one knows which way the plot went off scale. Since the actual value of the  $pH$  is also output, no information is lost. Figures 1, 2, and 3 show sample output for the titrations of two acids and a base, each with a different number of dissociation constants.

On a Univac Series 70/46 computer, only  $5\frac{1}{2}$  min cpu time was needed to titrate a total of 50 acids with from 1-4 dissociation constants each. The program requires 26,000 bytes (6,500 words) of core memory in the computer for execution.

Card decks, listings, input instructions, and test data are available from the author. An interactive version of this program written in Basic is also available.