

Simulação de uma Curva de Titulação Ácido Fraco — Base Forte

Fernando M. S. Silva Fernandes
Leonel A. T. P. Neves
Departamento de Química e CECUL
Faculdade de Ciências de Lisboa

1. Introdução

Um dos problemas que se coloca frequentemente na vida prática dum químico é a titulação dum ácido por uma base. Para além do aparato experimental necessário para a experiência, tem de saber-se qual o indicador apropriado para detectar o ponto final da neutralização que se pretende realizar.

Este problema é importante sob o ponto de vista experimental pois a detecção visual do ponto final numa titulação, além de ser uma prática corrente, não implica os erros experimentais que à primeira vista podem ser sugeridos por estar envolvido um aspecto subjectivo [1,2].

O problema resolve-se, experimentalmente, fazendo ensaios prévios com um sistema análogo ao que se pretende analisar, detectando as variações bruscas de pH no ponto de equivalência e, então, procurando numa tabela de indicadores o mais apropriado de acordo com o seu pK_i e a sua zona de viragem [2].

O objectivo deste trabalho é mostrar que esses ensaios prévios podem ser realizados por um microcomputador sem a necessidade de os executar experimentalmente. Para tal, simula-se a curva de titulação resolvendo as equações que dão o pH numa mistura de ácido fraco e de base forte para diferentes concentrações do ácido e da base na mistura. Uma vez determinado o pH no ponto de equivalência, pesquisa-se numa tabela o indicador mais apropriado.

Pode pensar-se que uma metodologia deste género conduz a uma situação semelhante à de já não se saber, em geral, como calcular uma raiz quadrada ou um logaritmo sem a ajuda dum calculadora. Todavia, a situação é diferente, pois o problema subjacente a uma titulação e escolha de indicadores é normalmente tratado a nível teórico com suficiente detalhe. Por outro lado, a possibilidade de gerar curvas de titulação no vídeo do microcomputador, para diferentes sistemas e situações, facilitará certamente a aprendizagem.

2. Equações Relevantes

Vamos supor que partimos dum certo volume, VA (em cm^3), dum solução aquosa de ácido fraco HA (o ácido acético, por exemplo) de concentração molar CA. Daqui podemos determinar o número de moles do ácido existentes no volume VA:

$$NA = VA * CA / 1000 \quad (1)$$

Pouco a pouco vamos deitando sobre o ácido um certo número de moles, NB, de base forte (o hidróxido de

sódio, por exemplo) processo que é controlado pela coordenada da titulação $f(i)$:

$$NB = f(i) * NA \quad (2)$$

Fazemos variar $f(i)$ de 0 a 2 com incrementos de 0.05 o que conduzirá a uma titulação com 41 passos; i variará de 0 a 40 correspondendo o passo 20 ao ponto de equivalência. A partir de NB e da concentração molar de base, CB, pode calcular-se o volume total da base adicionado ao fim de cada passo:

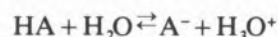
$$VB = 1000 * NB / CB \quad (3)$$

e, consequentemente, o volume total da mistura $VT = VA + VB$.

Daqui é imediato o cálculo das concentrações totais do ácido e da base na mistura ao fim de cada passo e supondo que ainda não reagiram: $AT = 1000 * NA / VT$ e $BT = 1000 * NB / VT$ respectivamente.

Como temos uma titulação com 41 passos teremos uma curva de titulação com 41 valores de pH que serão armazenados em $pH(i)$.

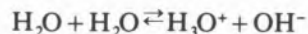
Antes do ponto de equivalência, isto é, enquanto $0 \leq f < 1$ temos a considerar os seguintes equilíbrios:



para o qual se define a constante de acidez:

$$KA = [A^-] [H_3O^+] / [HA] \quad (4)$$

e



para o qual se define o produto iónico da água

$$KW = [H_3O^+] [OH^-] \quad (5)$$

Como a base é forte considera-se completamente ionizada e os iões OH^- dela provenientes reagem com os H_3O^+ provenientes do ácido.

As espécies electricamente carregadas que existem na solução são: A^- , H_3O^+ , OH^- e Na^+ . É fácil concluir que:

$$AT = [HA]_{eq} + [A^-]_{eq} \quad (6)$$

Por outro lado, a condição de electroneutralidade é:

$$BT + [H_3O^+]_{eq} = [A^-]_{eq} + [OH^-]_{eq} \quad (7)$$

traduzindo o facto do número de cargas positivas ser igual ao número de cargas negativas.

As expressões (4), (5), (6) e (7) constituem as condições

do problema. A partir delas e por simples manipulação algébrica é fácil eliminar todas as concentrações $[]_{eq}$, excepto $[H_3O^+]_{eq}$. Tal eliminação conduz à seguinte equação cúbica (que será designada no algoritmo por APE):

$$[H_3O^+]^3 + (KA + BT) * [H_3O^+]^2 + (KA * BT - KA * AT - KW) * [H_3O^+] - KA * KW = 0 \quad (8)$$

em que $[H_3O^+]$ é a incógnita. Todas as outras quantidades que figuram na equação são determinadas a priori.

Resolvendo esta equação para diferentes valores de AT e BT pode calcular-se o pH da mistura ao longo do processo antes do ponto de equivalência.

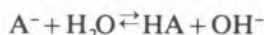
No ponto de equivalência e depois, isto é, quando $f \geq 1$ teremos:

a) quando $f = 1$ ($NB = NA$) formam-se NA moles de sal NaA; é o ponto de equivalência.

b) quando $f > 1$ continuarão a existir NA moles de sal e $NB - NA$ moles de base em excesso.

As concentrações do sal e da base em excesso serão $ST = 1000 * NA / VT$ e $BE = 1000 * (NB - NA) / VT$ respectivamente.

O sal, totalmente ionizado, sofrerá hidrólise e a base em excesso considera-se, também, totalmente ionizada. Os equilíbrios presentes serão:



para o qual se define a constante de hidrólise:

$$KH = [HA] [OH^-] / [A^-] = KW / KA \quad (9)$$

e novamente o equilíbrio da auto-ionização da água com a respectiva constante KW.

Temos a considerar, agora, os seguintes balanços:

$$ST = [A^-]_{eq} + [HA]_{eq} \quad (10)$$

e

$$BE + ST + [H_3O^+]_{eq} = [OH^-]_{eq} + [A^-]_{eq} \quad (11)$$

sendo esta última a condição de electroneutralidade.

Subtraindo membro a membro as equações anteriores obtém-se:

$$[HA]_{eq} = [OH^-]_{eq} - [H_3O^+]_{eq} - BE \quad (12)$$

As equações (9), (5), (11) e (12) constituem as condições do problema. De modo semelhante ao que se fez anteriormente é fácil eliminar entre elas todas as concentrações $[]_{eq}$, excepto $[H_3O^+]$, obtendo-se a seguinte equação cúbica (que será designada no algoritmo por PEDP):

$$[H_3O^+]^3 + (BE + ST + \frac{KW}{KH}) * [H_3O^+]^2 - (KW - BE * \frac{KW}{KH}) * [H_3O^+] - \frac{KW^2}{KH} = 0 \quad (13)$$

em que $[H_3O^+]$ é a incógnita. Todas as outras quantidades são também determinadas a priori como na equação (8). Como $KH = KW / KA$ a equação anterior pode ser rescrita na forma:

$$[H_3O^+]^3 + (BE + ST + KA) * [H_3O^+]^2 - (KW - BE * KA) * [H_3O^+] - KW * KA = 0 \quad (14)$$

Resolvendo esta equação para o valor de ST e para diferentes valores de BE pode calcular-se o pH da mistura ao longo do processo no ponto de equivalência e após o ponto de equivalência.

3. Método de Resolução das Equações

Embora haja métodos analíticos para resolver equações cúbicas [3] as fórmulas resolventes não são tão simples como para as equações do 2.º grau. Por outro lado, no caso das equações anteriores só estamos interessados na raiz real e positiva. Finalmente, se tivermos equações algébricas de grau igual ou superior a cinco não existem, em geral, fórmulas resolventes. É de toda a conveniência, então, que se estabeleçam métodos numéricos que nos permitam resolver equações de qualquer grau e que sejam simples de implementar num computador.

O método numérico que utilizamos para resolver as equações (8) e (13) é o método das bissecções sucessivas. Trata-se de um método sugestivo, simples de compreender e de implementar e, para o caso presente, suficiente rápido. Os leitores interessados encontrarão outros métodos, porventura mais expeditos, nas referências [4,5].

O método das bissecções sucessivas é um método iterativo que pressupõe o conhecimento prévio dum intervalo $[X1, X2]$ onde a raiz se encontre. A determinação desse intervalo pode fazer-se, por exemplo, por estudos gráficos prévios ou, então, a partir de considerações físicas que condicionem o problema.

No caso presente, vejamos que podemos determinar um intervalo por simples considerações físicas. Assim: a) Antes do ponto de equivalência a concentração $[H_3O^+]$ teórica mínima será 0, pois concentrações negativas não são fisicamente possíveis.

A concentração $[H_3O^+]$ teórica máxima será $AT + (KW)^{1/2}$ que corresponde à ionização total do ácido. Portanto:

$$X1 = 0 \quad e \quad X2 = AT + (KW)^{1/2} \quad (15)$$

No algoritmo fazemos a identificação $X = [H_3O^+]$.

b) No ponto de equivalência e depois é óbvio que:

$$X1 = 0 \quad e \quad X2 = 10^{-7} \quad (16)$$

Uma vez determinado o intervalo $[X1, X2]$ calcula-se o seu ponto médio:

$$XM = (X1 + X2) / 2 \quad (17)$$

e pesquisa-se em qual dos subintervalos $[X1, XM]$ e $[XM, X2]$ a raiz se encontra.

Se a raiz está em $[X1, XM]$ então o novo valor de X2 passará a ser XM e faremos $X2 = XM$. Senão a raiz encontra-se em $[XM, X2]$ e o novo valor de X1 passará a

ser XM e faremos $X1 = XM$. O processo é então repetido.

É evidente que pela repetição deste processo podemos aproximar-nos da raiz tanto quanto quisermos porque, em cada passo, reduzimos a metade o intervalo onde a raiz se encontra. Assim, 10 passos reduzirão o intervalo por um factor de 2^{10} ou seja cerca de 1000, enquanto 20 passos reduzirão o intervalo por um factor de 2^{20} ou seja cerca de 1 000 000.

O processo tem necessariamente de terminar e isso é feito logo que o comprimento do intervalo $|X1 - X2| < PE$ onde PE é um valor pré-estabelecido e que depende do erro que queiramos estabelecer para o método. No caso presente, uma boa escolha, que os leitores poderão comprovar, será $10^{-6} - 10^{-8}$ antes do ponto de equivalência e $10^{-11} - 10^{-14}$ no ponto de equivalência e depois.

Resta estabelecer um critério para decidir em que subintervalo a raiz se encontra após cada partição do intervalo $[X1, X2]$. Tal critério baseia-se no seguinte resultado da análise matemática [4]: se uma função $f(x)$ tem uma única raiz no intervalo $[X1, X2]$ então $f(X1) * f(X2) < 0$. Este resultado é intuitivo se repararmos que quando o gráfico duma função corta o eixo dos x num ponto (esse ponto será a raiz) a função muda de sinal.

Após cada partição do intervalo analisa-se o valor de $f(X1) * f(XM)$. Se for positivo então a raiz estará em $[XM, X2]$ e atribui-se o valor de XM a $X1$. Senão a raiz estará em $[X1, XM]$ e atribui-se XM a $X2$. É claro que, logicamente, deveremos considerar o caso do valor ser nulo o que corresponderá a XM ser a raiz.

No nosso caso, a função f é a função da expressão (8) que no algoritmo se designa por $APE(X)$ ou a função das expressões (13) ou (14) que no algoritmo se designa por $PEDP(X)$.

Quando o processo terminar, isto é, quando $|X1 - X2| < PE$ considera-se a raiz, que no algoritmo será designada por H , como o ponto médio do intervalo, valor que é afectado por um erro máximo de $PE/2$. Uma vez determinado o pH no ponto de equivalência procura-se numa tabela, com $NIND$ indicadores, o indicador cuja zona de viragem contenha o pH do ponto de equivalência. A tabela contém os nomes dos indicadores em $NOMEIND(i)$, os respectivos pK_i em $PKI(i)$ e os limites inferior e superior da zona de viragem para cada indicador em $LI(i)$ e $LS(i)$ respectivamente; i varia de 1 a $NIND$. Se $LI(i) < pH(20) < LS(i)$ então o indicador cujo nome está em $NOMEIND(i)$ será apropriado para a titulação considerada.

4. Algoritmo

Apresentamos seguidamente o algoritmo para a simulação da curva de titulação dum ácido fraco por uma base forte e para a escolha do indicador apropriado. O algoritmo é representado em notação algorítmica [6,7]. Esta notação é um meio-termo entre uma linguagem natural (o português, por exemplo) e uma linguagem de programação particular (o BASIC ou o PASCAL, por exemplo). A sua utilização liberta o leitor do conhecimento duma linguagem de programação particular e das ambiguidades e extensão duma linguagem natural. Por outro lado, a notação algorítmica destaca a lógica subjacente no algoritmo. Dado que não se trata duma linguagem, não tem regras sintáticas rígidas e os leitores são livres de representar as operações primitivas da forma que mais lhes agradar. A única condi-

ção é que a forma escolhida seja sucinta, clara e não tenha ambiguidades. Como a notação algorítmica é suficientemente clara não faremos quaisquer considerações detalhadas sobre ela. Estamos certos que os leitores se aperceberão das suas particularidades e vantagens ao longo da leitura do algoritmo. Este é constituído por uma parte principal da qual é chamada a subrotina $RAIZ$ que resolve as equações que referimos atrás. A subrotina e as funções encontram-se descritas após o algoritmo principal.

A codificação do algoritmo numa linguagem de programação particular, conduzindo ao programa, é um passo rotineiro que nada tem a ver com a lógica subjacente, mas apenas com as regras sintáticas da linguagem escolhida.

Posto isto, passemos ao algoritmo. Tudo o que se encontra entre chavetas são apenas comentários.

ALGORITMO PRINCIPAL

{Leitura do número e tabela de indicadores}.

Ler $NIND$

Para i variando de 1 a $NIND$ repetir

 Ler $NOMEIND(i)$, $PKI(i)$, $LI(i)$, $LS(i)$

{Leitura de variáveis}

Ler VA , CA , CB , KA , PE

$KW \leftarrow 1.0 \text{ E-}14$

$NA \leftarrow VA * CA / 1000$

Escrever VA , CA , CB , KA , KW , PE

{Antes do ponto de equivalência}

Para i variando de 0 a 19 repetir

$f(i) \leftarrow i * 5 / 100$

$NB \leftarrow f(i) * NA$

$VB \leftarrow 1000 * NB / CB$

$VT \leftarrow VA + VB$

$AT \leftarrow 1000 * NA / VT$

$BT \leftarrow 1000 * NB / VT$

 {Parâmetros de subrotina $RAIZ$ }

$X1 \leftarrow 0$

$X2 \leftarrow AT + \sqrt{KW}$

$BAND \leftarrow 0$ { $BAND = 0$ indica função $APE(X)$;
 $BAND = 1$ a função $PEDP(X)$ }

 CHAMAR SUBROTINA $RAIZ$

 Escrever AT , BT , H , $ERRO$, $f(i)$, $pH(i)$

{Ponto de equivalência e depois}

Ler PE

Para i variando de 20 a 40 repetir

$f(i) \leftarrow i * 5 / 100$

$NB \leftarrow f(i) * NA$

$VB \leftarrow 1000 * NB / CB$

$VT \leftarrow VA + VB$

$ST \leftarrow 1000 * NA / VT$

$BE \leftarrow 1000 * (NB - NA) / VT$

 {Parâmetros de subrotina $RAIZ$ }

$X1 \leftarrow 0$

$X2 \leftarrow 1.0 \text{ E-}7$

$BAND \leftarrow 1$

CHAMAR SUBROTINA RAIZ

Escrever ST, BE, H, ERRO, f(i), pH(i)

{Representação Gráfica dos valores Calculados}

Para i variando de 0 a 40 repetir

PONTO (f(i) * FH, pH(i) * FV)

{PONTO é uma rotina gráfica que escreve um pixel no ponto (x,y) do vídeo: FH e FV são os factores de escala}

{Escolha do Indicador}

SENT ← 0 {Se SENT permanecer com o valor 0 não há indicador que sirva}

Para i variando de 1 a NIND repetir

Se (LI (i) < pH (20) < LS (i) então

Escrever NOMEIND(i), PKI(i), LI(i), LS(i),
pH(20)
SENT ← 1

Se SENT = 0 então

Escrever "A lista dos indicadores não contém um indicador apropriado para a titulação"

FIM DO ALGORITMO PRINCIPAL

SUBROTINA RAIZ

Se BAND = 0 então

FX1 ← APE (X1)
FX2 ← APE (X2)

Senão

FX1 ← PEDP (X1)
FX2 ← PEDP (X2)

Se FX1 * FX2 ≥ 0 então

Se FX1 * FX2 = 0 então

Se FX1 = 0 então

H ← X1

Senão

H ← X2

Senão

Escrever "Erro nos limites"
Parar

Senão

Enquanto |X1 - X2| > PE repetir

XM ← (X1 + X2) / 2

Se BAND = 0 então

FX1 ← APE(X1)
FXM ← APE(XM)

Senão

FX1 ← PEDP (X1)
FXM ← PEDP (XM)

Se FXM ≠ 0 então

Se FXM * FX1 > 0 então

X1 ← XM

Senão

X2 ← XM

Senão

X1 ← XM
X2 ← XM

H ← (X1 + X2) / 2

ERRO ← PE / 2

pH(i) ← Log₁₀(H)

RETORNO

FUNÇÃO APE (X)

APE ← X * X * X + (KA + BT) * X * X + (KA *
BT - KA * AT - KW) * X - KA * KW

FUNÇÃO PEDP (X)

PEDP ← X * X * X + (BE + ST + KA) * X * X -
(KW - BE * KA) * X - KA * KW

Muitos microcomputadores com capacidade gráfica têm como primitiva, uma rotina semelhante à rotina PONTO do nosso algoritmo, bastando invocá-la no programa. Essa invocação é feita normalmente através da instrução PLOT (x,y). Contudo, se um microcomputador com capacidade gráfica não tiver, como primitiva, uma rotina semelhante a PONTO terá certamente outras primitivas gráficas a partir das quais o utilizador poderá construir uma rotina PONTO.

Queremos sublinhar que um algoritmo traduz um padrão de comportamento sem referência a um processador particular. Um programa, no entanto, tem especificações detalhadas com respeito a um processador lógico particular. Isto significa que os leitores que pretendam codificar o algoritmo anterior devem estar alertados para o facto de certos microcomputadores não aceitarem, por exemplo, variáveis indexadas com mais de um símbolo para os seus identificadores. No nosso algoritmo os valores de pH ao longo do processo são armazenados em pH(i). Alguns microcomputadores não aceitam este identificador pelo facto de ter dois símbolos. Nesse caso pode, por exemplo, substituir-se pH(i) por p(i).

Por outro lado, as primitivas:

Se (condição) então

| acção 1

Senão

| acção 2

e

Enquanto (condição) repetir

| acção

que utilizámos ao longo do algoritmo podem não ter correspondentes directos em algumas linguagens de programação. De facto, as estruturas IF/THEN/ELSE e WHILE não existem ainda em todas as versões das linguagens de alto nível. Todavia é possível construir estruturas semelhantes com as instruções IF/THEN e GOTO as quais existem em todas as linguagens mais comuns. Mas isto são problemas sintáticos das linguagens de programação que podem ser facilmente resolvidos com o manual da linguagem utilizada e se tivermos, à partida, um algoritmo representado numa notação algorítmica clara.

5. Resultados

Na tabela 1 apresentamos alguns dos resultados de simulação duma neutralização de 50 cm³ de uma solução aquosa de ácido acético 0.1 M por uma solução aquosa de hidróxido de sódio 0.1 M.

Tabela 1
Resultados da Simulação de uma Titulação de Ácido Acético 0.1 M com Hidróxido de Sódio 0.1 M

f	pH
0	2.87
0.1	3.79
0.2	4.13
0.3	4.36
0.4	4.56
0.5	4.73
0.6	4.91
0.7	5.10
0.8	5.33
0.9	5.67
1.0	8.75
1.1	11.76
1.2	12.05
1.3	12.20
1.4	12.32
1.5	12.39
1.6	12.41
1.7	12.50
1.8	12.54
1.95	12.59

Para o processo antes do ponto de equivalência assumimos um erro de iteração de 10⁻⁸ o que implicou cerca de 20 iterações por passo; no ponto de equivalência, e depois, assumimos um erro de 10⁻¹⁴ o que implicou cerca de 40 iterações por passo. No total, foram realizadas cerca de 1200 iterações.

O tempo de execução depende evidentemente do microcomputador e da linguagem utilizados. Codificando o algoritmo anterior em BASIC, o respectivo programa executado num microcomputador de pequeno porte (o

ZX Spectrum, por exemplo) consumirá cerca de 15 min para construir a curva de titulação, desenhá-la e escolher o indicador. Num microcomputador de 16 bits (o ai - M16 ou o QL, por exemplo) o tempo será substancialmente reduzido, em particular, se a codificação do algoritmo for feita em PASCAL: 2 min quando muito. Na Fig. 1 representa-se a cópia da curva de titulação desenhada no vídeo. O indicador escolhido foi a fenolftaleína como era de esperar.

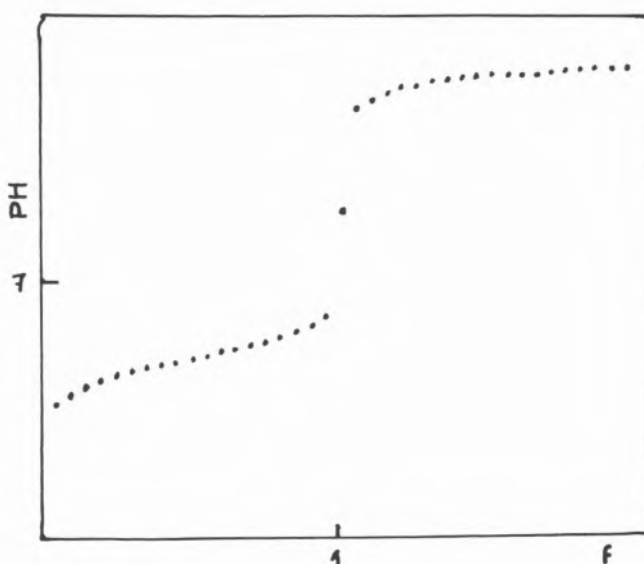


Fig. 1
Curva de uma Titulação Simulada do Ácido Acético 0.1 M com Hidróxido de Sódio 0.1 M

6. Conclusões

Os resultados obtidos mostram que é perfeitamente exequível utilizar um microcomputador para delinear uma titulação e escolher o indicador apropriado. Os erros envolvidos são comparáveis, ou mesmo inferiores, aos erros experimentais.

A única aproximação envolvida nas equações que utilizámos é a introdução de concentrações em vez de actividades. Tal aproximação, que é perfeitamente justificável se as concentrações forem suficientemente baixas pode, no entanto, ser facilmente remediada se as condições do problema o exigirem.

Por outro lado o algoritmo apresentado contempla apenas o caso de ácidos e bases monofuncionais. A sua adaptação ao caso geral de ácidos e bases polifuncionais não é difícil e deixamos essa adaptação ao critério dos leitores.

O uso da notação algorítmica parece-nos bastante importante pois facilita a compreensão do método e a codificação do algoritmo.

Finalmente, manifestamos aos leitores a nossa disponibilidade para o esclarecimento de quaisquer dúvidas que possam surgir na leitura deste trabalho.

REFERÊNCIAS

1. "University Chemistry", B. Mahan, 3.^a Edição, Addison-Wesley, 1975.
2. "Equilíbrio Químico e Reacções de Oxi-Redução", Fernando Fernandes, Emilia Carvalho e Helena Mendonça, Escôlar Editora, 1982.
3. "Mathematical Techniques in Chemistry", J. P. Dence; John Wiley e Sons, 1975.
4. "Elementary Numerical Analysis", Conte e de Boor; McGraw-Hill, 2.^a Edição, 1972.
5. "Numerical Methods in Chemistry", K. Jeffrey Johnson, Marcel Dekker, Inc., 1980.
6. "An Introduction to Data Structures with Applications", J. P. Tremblay e P. Sorenson, 2.^a Edição, McGraw-Hill, 1984.
7. "Introduction a la Programmation Structurée", Micro-Systems, 117, Maio-Junho, 1982.