

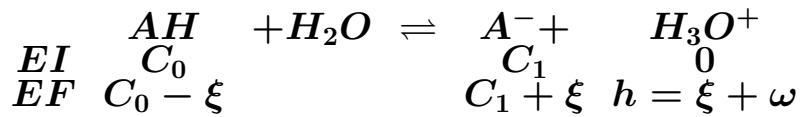
Calcul des pH

(texte original avec développements / démonstrations des formulations)

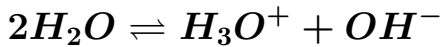
1.1. Cas d'un monoacide

Dans une solution aqueuse contenant un monoacide faible à la concentration C_0 , et sa base conjuguée à la concentration C_1 , il faut étudier les équilibres suivants :

- Réaction d'hydrolyse de l'acide : $K_A = 10^{-pK_A}$



- Autoprotolyse de l'eau : K_E



Pour déterminer le pH défini comme $-\log_{10}(h)$ de la solution, il faut trouver $h = \xi + \omega$ à partir de l'expression des constantes d'équilibre :

$$K_E = h\omega \quad \text{où} \quad h = [H_3O^+]([H_3O^+]_{tot}) \quad \text{et} \quad \omega = [OH^-]([H_3O^+]_{aq}) \quad \xi = [H_3O^+]_{ac}.$$

$$K_A = \frac{[A^-]h}{[AH^-]} = \frac{(C_1 + \xi)h}{(C_0 - \xi)}, \text{ en isolant } \xi, \text{ il apparaît } \xi = \frac{\frac{K_A C_0}{h} - C_1}{1 + \frac{K_A}{h}}$$

Il vient ensuite :

$$h = \xi + \omega = \frac{\frac{K_A C_0}{h} - C_1}{1 + \frac{K_A}{h}} + \frac{K_E}{h} \xrightarrow{*h}$$

$$h^2 = \frac{K_A C_0 - h C_1}{\frac{h + K_A}{h}} + K_E \Rightarrow h^3 + h^2 K_A = h K_A C_0 - h^2 C_1 + h K_E + K_A K_E$$

Soit :

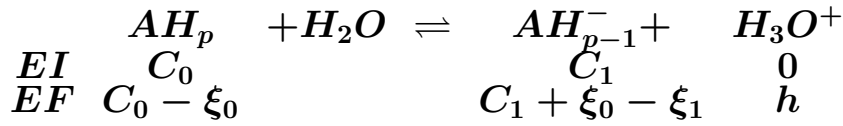
$$h^3 + (K_A + C_1)h^2 - (K_E + K_A C_0)h - K_A K_E = 0$$

L'expression obtenue est déjà relativement compliquée, c'est pourquoi, en pratique la méthode de la réaction prépondérante s'impose. Néanmoins, ce type de résolution ne pose pas de problème à Python (la généralisation de la recherche du pH d'un mélange quelconque est étudiée dans la suite). Évidemment l'expression obtenue est bien plus "indigeste" encore que l'expression polynomiale de degré trois du cas simple ci-dessus.

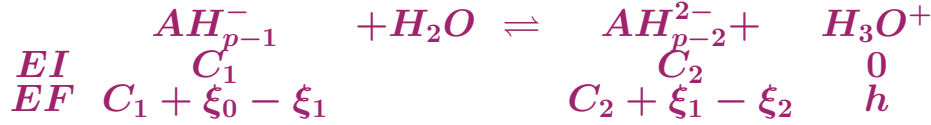
1.2. Cas d'un polyacide

Soit un polyacide AH_p comportant p acidités, le calcul du pH consiste à considérer les équilibres suivants :

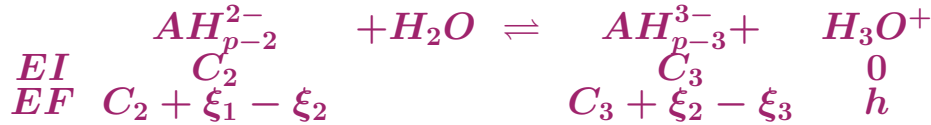
- 1er équilibre : constante : K_1



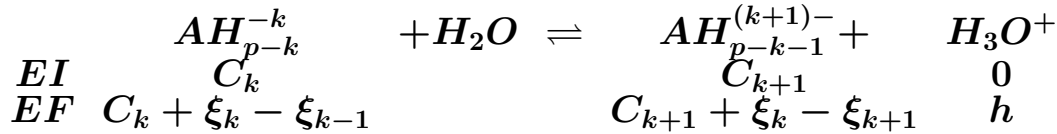
- 2ème équilibre : constante : K_2



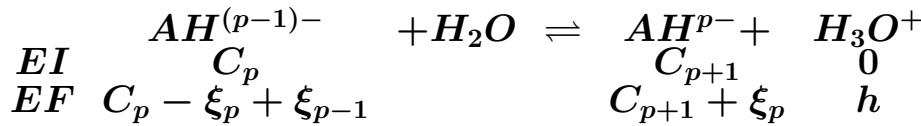
- 3ème équilibre : constante : K_3



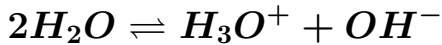
- $k + 1^{\text{ème}}$ équilibre : constante : K_{k+1}



- $p^{\text{ème}}$ équilibre : constante : K_p



Il faut également rajouter l'autoprotolyse de l'eau selon l'équilibre :



La concentration en ions H_3O^+ vaut $h = \sum_{k=0}^p \xi_k + \frac{K_e}{h} = 10^{-pH}$

Or, on constate que $\sum_{k=0}^p \xi_k = \sum_{k=0}^p (k+1) \left([AH_{p-k}^{-k}] - C_K \right)$

DEMONSTRATION

C_k étant la concentration du réactif avant dissociation et $[AH_{p-k}^{-k}]$ la concentration finale,

On a par les 3 premières réactions, après avoir mis en évidence les ξ_i et effectué leur sommation :

$$(1) C_0 - (C_0 - \xi_0) = C_0 - [AH_p] = \xi_0$$

$$(2) C_1 - (C_1 + \xi_0 - \xi_1) = C_1 - [AH_{p-1}^-] = -\xi_0 + \xi_1 \Rightarrow C_1 - [AH_{p-1}^-] + \overbrace{C_0 - [AH_p]}^{\xi_0} = \xi_1$$

$$C_2 - (C_2 + \xi_1 - \xi_2) = C_2 - [AH_{p-2}^{2-}] = -\xi_1 + \xi_2 \Rightarrow$$

$$(3) C_2 - [AH_{p-2}^{2-}] + \overbrace{C_1 - [AH_{p-1}^-]}^{\xi_1 - \xi_0} + \overbrace{C_0 - [AH_p]}^{\xi_0} = \xi_2$$

$$3(C_0 - [AH_p]) + 2(C_1 - [AH_{p-1}^-]) + 1(C_2 - [AH_{p-2}^{2-}]) = \xi_0 + \xi_1 + \xi_2$$

Or, on constate qu'en inversant les coefficients et l'ordre de la différence, on obtiendrait le même résultat, lorsque i tend vers p :

En effet :

$$([AH_p] - C_0) + 2([AH_{p-1}^-] - C_1) + 3([AH_{p-2}^{2-}] - C_2) + 4([AH_{p-3}^{3-}] - C_3) \dots = -\xi_0 + 2(\xi_0 - \xi_1) + 3(\xi_1 - \xi_2) + 4(\xi_2 - \xi_3) \dots$$

$$= \cancel{-\xi_0} + 2\xi_0 - \cancel{2\xi_1} + 3\xi_1 - \cancel{3\xi_2} + 4\xi_2 - 3\xi_3 = \xi_0 + \xi_1 + \xi_2 + \xi_3 \dots$$

En formulant par une sommation en série, de k allant de 0 à p,

$$(0 + 1)([AH_p] - C_0) + (1 + 1)(C_1 - [AH_{p-1}^-]) + (2 + 1)([AH_{p-2}^{2-}] - C_2) \dots = \sum_{k=0}^p \xi_k$$

$$\text{On retrouve bien la formule générale : } \sum_{k=0}^p \xi_k = \sum_{k=0}^p (k + 1) \left([AH_{p-k}^{-k}] - C_k \right) \quad C.Q.F.D.$$

$$\text{En traduisant la conservation de l'entité A, il vient } \sum_{k=0}^p [AH_{p-k}^{-k}] = \sum_{k=0}^p C_k$$

(Principe de la conservation de la concentration totale de l'acide avant et après dissociation)

De plus, l'écriture des constantes d'acidité donne :

$$K_1 = \frac{h[AH_{p-1}^-]}{[AH_p]}, \dots \quad K_{h+1} = \frac{h[AH_{p-k-1}^{(k+1)-}]}{[AH_{p-k}^{k-}]}, \dots \quad K_p = \frac{h[A^{p-}]}{[AH^{(p-1)-}]}$$

L'expression des différentes concentrations devient :

$$(4)[AH_p] = C \frac{1}{1 + \frac{K_1}{h} + \frac{K_1 K_2}{h^2} + \dots + \frac{\prod_{k=1}^p K_k}{h^p}}, \dots$$

$$(5)[AH_{p-k}^{k-}] = C \frac{\frac{\prod_{i=1}^k K_i}{h^k}}{1 + \frac{K_1}{h} + \frac{K_1 K_2}{h^2} + \dots + \frac{\prod_{k=1}^p K_k}{h^p}} \quad \text{càd} \quad [AH_{p-k}^{k-}] = C \frac{\frac{\prod_{i=1}^k K_i}{h^k}}{1 + \sum_{k=1}^p \frac{\prod_{i=1}^k K_i}{h^k}}$$

Avec la convention : $\prod_{i=1}^0 K_i = 1$

DEMONSTRATION

La concentration totale en acide est donnée par la somme de celui-ci sous toutes ses formes :

$$C = [AH_p] + [AH_{p-1}^-] + [AH_{p-2}^{2-}] + \dots = [AH_p] + \frac{K_1[AH_p]}{h} + \frac{K_2 \overbrace{[AH_{p-1}^-]}^{\frac{K_1[AH_p]}{h}}}{h} + \dots$$

$$C = [AH_p] + \frac{K_1[AH_p]}{h} + \frac{K_1 K_2[AH_p]}{h^2} + \dots = [AH_p] \left(1 + \frac{K_1}{h} + \frac{K_1 K_2}{h^2} + \dots \right) \Rightarrow$$

$$(4)[AH_p] = \frac{C}{1 + \frac{K_1}{h} + \frac{K_1 K_2}{h^2} + \dots + \frac{\prod_{k=1}^p K_k}{h^p}} \quad \text{C.Q.F.D.}$$

En effectuant le produit des constantes d'équilibre, on a également par simplification :

$$\prod_{i=1}^k K_i = K_1 * K_2 * K_3 \dots = \frac{h[\cancel{AH_{p-1}^-}]}{[AH_p]} * \frac{h[\cancel{AH_{p-2}^{2-}}]}{[\cancel{AH_{p-1}^-}]} * \frac{h[\cancel{AH_{p-3}^{3-}}]}{[\cancel{AH_{p-2}^{2-}}]} \dots * h[AH_{p-k}^{k-}] =$$

$$\prod_{i=1}^k K_i = h^k \frac{[AH_{p-k}^{k-}]}{[AH_p]} \Rightarrow [AH_{p-k}^{k-}] = \frac{\prod_{i=1}^k K_i [AH_p]}{h^k} \stackrel{(4)}{=} \frac{\prod_{i=1}^k K_i}{h^k} * \frac{C}{1 + \frac{K_1}{h} + \dots + \frac{\prod_{k=1}^p K_k}{h^p}}$$

$$(5)[AH_{p-k}^{k-}] = C * \frac{\frac{\prod_{i=1}^k K_i}{h^k}}{1 + \frac{K_1}{h} + \frac{K_1 K_2}{h^2} \dots + \frac{\prod_{k=1}^p K_k}{h^p}} = C \frac{\frac{\prod_{i=1}^k K_i}{h^k}}{1 + \sum_{k=1}^p \frac{\prod_{i=1}^k K_i}{h^k}} \quad \text{C.Q.F.D.}$$

$$\text{Or, } \sum_{k=0}^p \xi_k = \sum_{k=0}^p (k+1) \left([AH_{p-k}^{k-}] - C_K \right) \quad \text{et} \quad h = \sum_{k=0}^p \xi_k + \frac{K_e}{h} = 10^{-pH}$$

$$\text{Donc, } h = \sum_{k=0}^p (k+1) \left([AH_{p-k}^{k-}] - C_K \right) + \frac{K_e}{h}$$

$$\text{En posant } f : h \rightarrow h - \sum_{k=0}^p \left((k+1) \left(C \frac{\frac{\prod_{j=1}^k K_j}{h^k}}{1 + \sum_{k=1}^p \frac{\prod_{j=1}^k K_j}{h^k}} - C_k \right) \right) - \frac{K_e}{h}$$

Le calcul de pH revient à chercher les zéros de la fonction f.

j étant ici l'indice de la réaction de dissociation (nombre de K) de l'acide, son étendue est d'une unité inférieure à k qui est l'indice de composés acides

1.3. Cas d'un mélange quelconque

Dans le cas où un mélange est composé de n acides notés $A_i H_{p_i}$, où p_i est le nombre d'acidité du $i^{\text{ème}}$ acide, il faut sommer sur tous les acides, ce qui conduit alors pour déterminer le pH, à chercher les zéros de la fonction suivante :

$$\left[f : h \rightarrow h - \sum_{j=1}^N \left(\sum_{k=0}^{p_j} \left((k+1) \left(\sum_{k=0}^{p_j} C_{k,j} \frac{\prod_{i=1}^k K_{i,j}}{h^k} - C_{k,j} \right) \right) \right) - \frac{K_e}{h} \right]$$

Sans démonstration, on a vu que j étant ici l'indice de la réaction de dissociation (nombre de K) pour un acide donné, son étendue est d'une unité inférieure à k qui est l'indice de composés acides, le nouvel indice i est l'indice de l'acide, il est évident que l'on peut considérer indépendamment les séries de différentes réactions acides et les sommer pour obtenir l'équation ci-dessous. Cependant, les indices semblent être inversés, la formule exacte serait :

$$f : h \rightarrow h - \sum_{i=1}^N \left(\sum_{k=0}^{p_j} \left((k+1) \left(\sum_{k=0}^{p_j} C_{i,j} \frac{\prod_{j=1}^k K_{i,j}}{h^k} - C_{i,j} \right) \right) \right) - \frac{K_e}{h}$$

$K_{i,j}$ est la $j^{\text{ème}}$ constante du $i^{\text{ème}}$ acide et $C_{i,j}$ est la $j^{\text{ème}}$ concentration initiale du $i^{\text{ème}}$ acide. Les pK_A (sachant que $K_A = 10^{-pK_A}$) et les concentrations sont saisis dans les lignes des matrices (la sommation étant plus facile par ligne).

2. Exemples de calculs

2.1. Présentation

soit le mélange suivant :

■ Acide citrique :

- $[H_3A] = C_{1,1} = 0.1[mol.L^{-1}]$; $pK_{1,1} = 3.1$
- $[H_2A^-] = C_{2,1} = 0$; $pK_{2,1} = 4.8$
- $[HA^{2-}] = C_{3,1} = 0$; $pK_{3,1} = 6.4$
- $[A^{3-}] = C_{4,1} = 0$, dans ce cas : $p_1 = 3$.

■ Acide sulfureux :

- $[H_2SO_3] = C_{1,2} = 0.1mol.L^{-1}$; $pK_{1,2} = 1.8$
- $[HSO_3^-] = C_{2,2} = 0$; $pK_{2,2} = 7.2$
- $[SO_3^{2-}] = C_{3,2} = 0$, dans ce cas : $p_2 = 2$.

■ Ammonium :

- $[NH_4^+] = C_{1,3} = 0$; $pK_{1,3} = 7.2$
- $[NH_3] = C_{2,3} = 0.6mol.L^{-1}$, et $p_3 = 1$.

2.2. Création des matrices de données et opérations

Les données peuvent être groupées sous forme de tableaux (matrices) :

$$K = \begin{pmatrix} K_{1,1} & K_{2,1} & K_{3,1} \\ K_{1,2} & K_{2,2} & 0 \\ K_{1,3} & 0 & 0 \end{pmatrix} = \begin{pmatrix} 10^{-3.1} & 10^{-4.8} & 10^{-6.4} \\ 10^{-1.8} & 10^{-7.2} & 0 \\ 10^{-9.2} & 0 & 0 \end{pmatrix}$$

$$C = \begin{pmatrix} C_{1,1} & 0 & 0 & 0 \\ C_{1,2} & 0 & 0 & 0 \\ 0 & C_{2,3} & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0.1 & 0 & 0 & 0 \\ 0.1 & 0 & 0 & 0 \\ 0 & 0.6 & 0 & 0 \end{pmatrix}$$

Les premières lignes des matrices K et C correspondent aux données de l'acide citrique H_3A , les secondes colonnes à H_2SO_3 et les troisièmes à NH_4^+ . Pour arriver à l'expression de la fonction f établie précédemment, il faut notamment effectuer la suite des opérations suivantes :

$$C = \begin{pmatrix} C_{1,1} & 0 & 0 & 0 \\ C_{1,2} & 0 & 0 & 0 \\ 0 & C_{2,3} & 0 & 0 \end{pmatrix} \longrightarrow C_0 = \begin{pmatrix} C_1 & C_1 & C_1 & C_1 \\ C_2 & C_2 & C_2 & C_2 \\ C_3 & C_3 & C_3 & C_3 \end{pmatrix} \text{ où } C_k = \sum_{i=0}^4 C_{i,k}$$

$$K = \begin{pmatrix} K_{1,1} & K_{2,1} & K_{3,1} \\ K_{1,2} & K_{2,2} & 0 \\ K_{1,3} & 0 & 0 \end{pmatrix} \xrightarrow{\frac{K}{h}} \begin{pmatrix} \frac{K_{1,1}}{h} & \frac{K_{2,1}}{h} & \frac{K_{3,1}}{h} \\ \frac{K_{1,2}}{h} & \frac{K_{2,2}}{h} & 0 \\ \frac{K_{1,3}}{h} & 0 & 0 \end{pmatrix} \xrightarrow{\text{concaténation}} \begin{pmatrix} 1 & \frac{K_{1,1}}{h} & \frac{K_{2,1}}{h} & \frac{K_{3,1}}{h} \\ 1 & \frac{K_{1,2}}{h} & \frac{K_{2,2}}{h} & 0 \\ 1 & \frac{K_{1,3}}{h} & 0 & 0 \end{pmatrix}$$

$$\begin{pmatrix} \frac{K_{1,1}}{h} & \frac{K_{2,1}}{h} & \frac{K_{3,1}}{h} \\ \frac{K_{1,2}}{h} & \frac{K_{2,2}}{h} & 0 \\ \frac{K_{1,3}}{h} & 0 & 0 \end{pmatrix} \xrightarrow{\text{produit cumulé}} \alpha_{num} \begin{pmatrix} 1 & \frac{K_{1,1}}{h} & \frac{K_{1,1}K_{2,1}}{h^2} & \frac{K_{1,1}K_{2,1}K_{3,1}}{h^3} \\ 1 & \frac{K_{1,2}}{h} & \frac{K_{1,2}K_{2,2}}{h^2} & 0 \\ 1 & \frac{K_{1,3}}{h} & 0 & 0 \end{pmatrix}$$

Puis on obtient le tableau (α_{num}) dont les éléments de la $i^{\text{ème}}$ ligne sont la somme de la $i^{\text{ème}}$ ligne

$$\text{de } \alpha_{num}, \text{ c'est-à-dire : } 1 + \sum_{k=1}^{p_i} \frac{\prod_{j=1}^k K_{j,i}}{h^k}$$

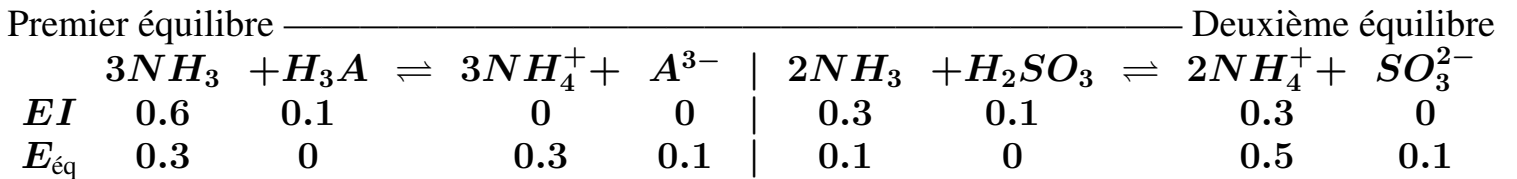
Ensuite la fonction f s'en déduit facilement à partir de la fonction sum. La détermination de la racine de f peut se réaliser par dichotomie dont le principe étudier plus en détails dans le chapitre sur la détermination des zéros d'une fonction.

2.3. Résultat numérique

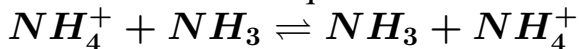
Le résultat numérique fourni par python est . pH = 8.53. La précision d'un pHmètre n'excède pas le premier chiffre après la virgule. Le résultat est donc approché à pH = 8.5.

2.4. Vérification avec la méthode de la réaction prépondérante (R.P.)

La bas étant en excès, il suffit de considérer les R.P. suivantes qui mènent au système équivalent ci-dessous.



R.P. de contrôle : équilibre indifférent



$$\text{donc, } pH = pK_A + \log \frac{NH_3}{NH_4^+} = 9.2 - \log 5 \approx 8.5 > pK_{1,3} + 1 \text{ et } pK_{2,2} + 1,$$

les réactions d'hydrolyse des autres bases sont négligeables. Ce qui conduit bien au même pH.

3. Codes

```
def fzero (C,K,h):
    Ke= 1e-14
    Nligne=len(C) # correspond aux nbr d'acides 3, dont chacun réagit avec H2O , comportant 4 réactifs
    Ncolonne= len(C[0]) # C a 4 valeurs (les 4 réactifs, deux formes acides, h2O & H3O+) C[0]=[0.1,0,0,0]
    # créat. de la matrice C0, initialisat. de la somme ,
    # calcul de la s. sur les lignes, ajout à la liste (matrice)
    C0=[]
    for liste in C:
        s=0
        for val in liste:
            s+=val
        C0.append([s]*Ncolonne)
    # calcul de la matrice K/h
    Ksurh=[[val/h for val in Kliste]for Kliste in K]
    # calcul de la matrice alphanum, lecture sur les lignes, ajout de 1 sur les lignes ,
    # initialisation du produit, produit sur les lignes
    # produit en place, ajout à la ligne de la matrice, ajout de la ligne
    alphanum=[]
    for liste in Ksurh:
        ligne=[1]
        p=1
        for val in liste:
            p*=val
            ligne.append(p)
        alphanum.append(ligne)
    # calcul de la matrice alphadénom, initialisation de la somme, somme sur les lignes
    alphadenom=[]
    for liste in alphanum:
        s=0
        for val in liste:
            s+=val
        alphadenom.append([s]*Ncolonne)
    # opération sur les 3 tableaux, multiplication et division
    # avec création d'itération sur 1 liste de tuples (zip)
    conc=[]
    for listel1,listel2,listel3 in zip(C0, alphanum,alphadenom):
        ligne=[]
        for vall, val2, val3 in zip(listel1,listel2,listel3):
            ligne.append(vall*val2/(1.*val3))
        conc.append(ligne)
    # Création d'un tableau de ligne 1,2,3..
    cumsum=[[1+j for j in range(Ncolonne)]]
    for i in range(Nligne):
        ksi_table=[]
    # Opération multiplication et différence sur les trois tableaux
    for listel1,listel2,listel3 in zip(cumsum,conc,C):
        ligne=[]
        for vall, val2, val3 in zip(listel1,listel2,listel3):
            ligne.append(vall*(val2-val3))
        ksi_table.append(ligne)
    ksi= 0
    for liste in ksi_table:
        ksi=ksi+sum(liste)
    return ksi-Ke/h-h
    # retour de la fonction
# Fonction permettant le calcul du pH par la donnée des matrices C et K.
# Recherche le résultat entre pH = -3 et 18
# avec une précision de 10^-2.
def calculpH(C,K):
    a=-3.
    b=18.
    epsi=1e-2
    # Boucle while tant que la précision n'est pas atteinte .
    # Algorithme de dichotomie: division de l'intervalle par 2 h = 10^-pH
```

```

while (abs(b-a)>epsi):
    c=(a+b)/2.
    if (fzero(C,K,10**(-c))*fzero(C,K,10**(-a))>0:
        a=c
    else:
        b=c
return c
K=[[10**-3.1,10**-4.8,10**-6.4],[10**-1.8,10**-7.2,0],[10**-9.2,0,0]]
C=[[0.1,0,0,0],[0.1,0,0,0],[0,0.6,0,0]]
print (calculpH(C,K))

```

Code version optimisée

```

def fzero (C,K,h):
    Ke= 1e-14
    Nligne=len(C)
    Ncolonne= len(C[0])
    alphanum=[]
    for liste in K:
        ligne=[1]
        p=0
    k=1
        for val in liste:
            p +=val
            ligne.append(p7h**k)
        k +=1
        alphanum.append(ligne)
    print alphanum
    conc=[]
    for liste1,liste2 in zip(C, alphanum):
        ligne=[]
    C0 = [sum(C)]*Ncolonne
    s = sum(liste2)
        for val1,val2 in zip(C0,liste2):
            ligne.append(val1*val2/(s))
        conc.append(ligne)
    cumsum=[[1+j for j in range(Ncolonne)]]
    for i in range(Nligne)]
    ksi_table=[]
    for liste1,liste2,liste3 in zip(cumsum,conc,C):
        ligne=[]
        for val1,val2,val3 in zip(liste1,liste2,liste3):
            ligne.append(val1*(val2-val3))
        ksi_table.append(ligne)
    ksi=sum([sum(liste) for liste in ksi_table])
    return ksi-Ke/h-h

```

Le code suivant réalise les mêmes calculs mais utilise largement les méthodes de la bibliothèque `pylab`. Il s'agit d'une émulation de `MATLAB`. Son avantage est d'être plus rapide et plus stable vis-à-vis de la somme (surtout lors de l'ajout de termes négligeables comme les concentrations ou les constantes d'acidité).

```

import matplotlib.pyplot as plt
import numpy as np
from pylab import*
def fzero (C,K,h):
    # Ecriture des données en colonnes, façon Matlab, par transposition
    C=array(C).T
    K=array(K).T
    Ke= 1e-14

    # Détermination des dimensions de la matrice C, Etablissement de la formule de la fonction f
    Nligne=size(C,axis=0)
    Ncolonne= size(C,axis=1)
    C0=dot(ones((Nligne,1)),array([sum(C,axis=0)]))

```

```

ConCat=concatenate((one((1,Ncolonne)),K/h),axis=0)
alphanum=cumprod(ConCat,axis=0)
alphadenom=dot(ones((Nligne,1)), $\frac{1}{C}$ 
array([sum(alphanum,axis=0)]))
conc = C0* alphanum/ alphadenom
ksi=sum(cumsum(ones((Nligne,Ncolonne)), $\frac{1}{C}$ 
axis=0)*(conc-C))
return ksi+Ke/h-h

```
