

Courbe de Prédominance d'un Acide

```
<sxh python; title : courbe_predominance_acide.py> #!/usr/bin/env python # -*- coding: utf-8 -*- #
travail de KH, ba2 chimie 2012-2013
```

```
# Courbe de Prédominance d'un Acide # from math import * import matplotlib.pyplot as plt from
Tkinter import *
```

```
def equations (n,C,a,b,c,d,e,f,g,pKa1,pKa2,pKa3): #initialiser (valeur 0)
```

```
    pH = 0. #pH de départ
    if n==3:
        Ka1=10**(-pKa1)
        Ka2=10**(-pKa2)
        Ka3=10**(-pKa3)
```

```
        while pH < 14: #pH va jusque 14
```

```
            H = 10**(-pH) #équation pour avoir les concentrations en H3O+ aux
différents pH demandé
            O = (10**(-14)) / H #équation pour avoir les concentrations en OH-
aux différents pH demandé
            K = 1 + H/Ka3 + (H**2)/(Ka2*Ka3) + (H**3)/(Ka1*Ka2*Ka3) #équation
qui détermine la valeur de alpha H
            a0= 1/K # équation pour avoir alpha0
            a1=a0*(H/Ka3) # équation pour avoir alpha1
            a2=a1*(H/Ka2) # équation pour avoir alpha2
            a3=a2*(H/Ka1) # équation pour avoir alpha3
            H3Y = a3 * C # équation pour avoir la [H3Y]
            H2Y = a2 * C # équation pour avoir la [H2Y-]
            HY2 = a1 * C # équation pour avoir la [HY2-]
            Y3 = a0 * C # équation pour avoir la [Y3-]
```

```
# listes auxquels nous ajouterons les différentes valeurs calculées au dessus pour chaque pH
demandé :
```

```
    a.append(pH)
    b.append (H)
    c.append (O)
    d.append (H3Y)
    e.append (H2Y)
    f.append (HY2)
    g.append (Y3)
    pH = pH + 0.1
```

```
elif n==2:
    Ka1=10**(-pKa1)
    Ka2=10**(-pKa2)
```

```
        while pH < 14:
```

```
H = 10**(-pH) #équation pour avoir les concentration en H3O+ aux
différents pH demandé
O = (10**(-14)) / H # équation pour avoir les concentration en OH
- aux différents pH demandé
K = 1 + H/Ka2 + (H**2)/(Ka2*Ka1) #équation pour avoir la valeur
de alpha H
a0 = 1/K #équation pour avoir alpha 0
a1= a0*(H/Ka1) #équation pour avoir alpha 1
a2 = a1*(H/Ka1) #équation pour avoir alpha 2
H2Y = a2*C #équation pour avoir la [H2Y]
HY=a1*C #équation pour avoir la [HY-]
Y2=a0*C # équation pour avoir la [Y2-]
```

listes auxquels nous ajouterons les différentes valeurs calculées au dessus pour chaque pH demandé :

```
a.append(pH)
b.append(H)
c.append(O)
d.append(H2Y)
e.append(HY)
f.append(Y2)
pH = pH + 0.1
elif n==1:
    Ka1=10**(-pKa1)
    while pH < 14 :
        H = 10**(-pH) #équation pour avoir les concentration en H3O+ aux
différents pH demandé
        O = (10**(-14)) / H # équation pour avoir les concentration en
OH - aux différents pH demandé
        K = 1 + H/Ka1 #équation pour avoir la valeur de alpha H
        a0 = 1/K #équation pour avoir alpha 0
        a1= a0*(H/Ka1) #équation pour avoir alpha 1
        HY= a1*C #équation pour avoir la [HY]
        Y=a0*C #équation pour avoir la [Y-]
```

listes auxquels nous ajouterons les différentes valeurs calculées au dessus pour chaque pH demandé :

```
a.append(pH)
b.append (H)
c.append (O)
d.append (HY)
e.append (Y)
pH = pH + 0.1
```

#2eme fenêtré dans laquelle on rentrera les paramètres propre à la molécule étudiée. def fenetre2():

```
root2 = Tk()
```

```
n = float (champn.get ()) # transformation en nombre flottant de la valeur
entrée pour n
```

```
txtn = Label(root2, text ='pKa1 : ', fg = 'red')
txtn.grid(row=2)
champ1 = Entry(root2) #demander le pKa1
champ1.grid(row=3)      # ligne à laquelle apparaîtra le champs ou l'on
donnera le pKa1
if n > 1 :              #paramètre demander si la molécule possède 2 H+
```

```
txt2 = Label(root2, text ='pKa2 : ', fg = 'red')
txt2.grid(row=4)
champ2 = Entry (root2) #demande le pKa2
champ2.grid(row=5) # ligne à laquelle apparaîtra le champs ou l'on
donnera le pKa2
if n == 3 :            #paramètre demander si la molécule possède 3
H+
    txt3 = Label(root2, text ='pKa3 : ', fg = 'red')
    txt3.grid(row=6)
    champ3 = Entry (root2) #demande le pKa3
    champ3.grid(row=7) #ligne à laquelle apparaîtra le champs dans
lequel on donnera le pKa3
```

```
def graphe():
    a=[]
    b=[]
    c=[]
    d=[]
    e=[]
    f=[]
    g=[]
    n = float(champn.get())
    C = float(champc.get())
    pKa1 = float(champ1.get())
    pKa2 = 0
    pKa3 = 0
    if n > 1:
        pKa2 = float (champ2.get())
        if n == 3:
            pKa3 = float (champ3.get())
    equations(n,C,a,b,c,d,e,f,g,pKa1 = pKa1, pKa2 = pKa2, pKa3 = pKa3)
    plt.xlabel("pH")      #nom donné à l'axe des abscisses
    plt.ylabel("Concentrations") #nom donné à l'axe des ordonnées
```

les différentes droites qui seront présentes sur le graphique semilog :

```
c1 = plt.semilogy(a,b,'r') #courbe donnant le concentration en H3O+
en fonction du pH
c2 = plt.semilogy(a,c, 'b') #courbe donnant le concentration en OH- en
fonction du pH
c3 = plt.semilogy(a,e,'g') #courbe donnant le concentration en l'acide
```

```
qui à tout ces H+ en fonction du pH
    c4 = plt.semilogy(a,d,'m')
    if len(f)>1:
        c5 = plt.semilogy(a,f,'c') #courbe donnant le concentration en
l'acide qui a perdu 2H+ en fonction du pH
    if len(g)>1:
        c6 = plt.semilogy(a,g,'y') #courbe donnant le concentration en
l'acide qui a perdu 3H+ en fonction du pH
    plt.title("Courbe representant la predominance")
    plt.show()
    b2=Button(root2,text="Tracer le Graphique",command=graphe, bg = 'black',
fg = 'red') #bouton qui tracera la courbe de prédominance de l'acide
choisit.
    b2.grid(row=8)          #ligne à laquelle se situe le bouton2
```

```
    bfin=Button(root2,text="Quitter",command=root2.quit, bg = 'green', fg =
'blue')
    bfin.grid(row=9)          #ligne à laquelle se trouve le bouton de fin.
```

```
root2.mainloop()
```

```
# éliminer la fenêtre :
root2.destroy()
```

```
#fenêtre n°1 dans laquelle on donnera le nombre de proton de la molécule
afin de pouvoir donner les valeurs nécessaires à ceux-ci.
```

```
root1= Tk()
```

```
txtn = Label(root1, text = "Combien de proton peut perdre la molecule ?", fg = 'red')
txtn.grid(row=0) champn=Entry(root1) #demande le nombre de H+ de la molécule (1, 2 ou 3)
champn.grid(row=1) #ligne à laquelle apparaîtra le champ à compléter
```

```
txtc = Label(root1, text = "Concentration (mol/L) :", fg = 'orange') txtc.grid(row=2) champc =
Entry(root1) #demande la concentration de la molécule. champc.grid(row=3) #ligne à laquelle
apparaîtra le champ où l'on donnera la concentration
```

```
b1 = Button(root1, text = 'Valeurs des pKa ?',command = fenetre2,fg = 'red', bg = 'black') #bouton
qui ouvrira la fenêtre n°2 afin de rentrer les autres paramètres.
```

```
b1.grid(row = 4)
```

```
b2 = Button(root1, text = 'Quitter', command = root1.quit,fg = 'red', bg = 'black') b2.grid (row = 5)
root1.mainloop()
```

```
# éliminer la fenêtre : root1.destroy()
```

```
#Référence:
```

```
# http://w3.umons.ac.be/perso/Villers.Didier/wiki/doku.php?id=floss:python #
```

<http://www.siteduzero.com/informatique/python/tutoriels> #
<https://moodle.umons.ac.be/course/view.php?id=107> #
http://www.inforef.be/swi/download/apprendre_python3.pdf

</sxh>

From:
<https://dvillers.umons.ac.be/wiki/> - **Didier Villers, UMONS - wiki**

Permanent link:
https://dvillers.umons.ac.be/wiki/teaching:progappchim:courbe_predominance_acide_2013

Last update: **2013/11/28 14:18**

