

Activité 8

A la recherche du zéro

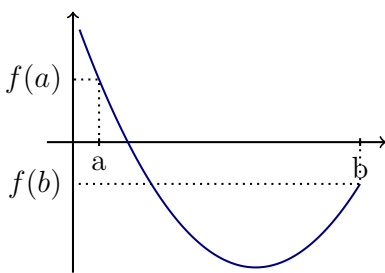
1

Méthodes

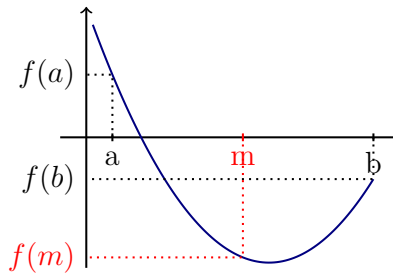
1.1

Méthode de dichotomie

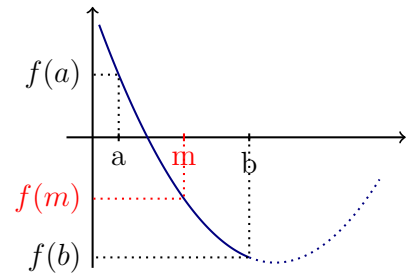
C'est une méthode relativement simple et efficace pour trouver l'unique zéro d'une fonction $f(x)$ sur un intervalle $[a, b]$ donné avec une précision eps souhaitée.



La fonction ne s'annulant qu'une seule fois sur l'intervalle d'étude, on doit avoir $f(a) \times f(b) < 0$. Considérons le cas de figure ci-contre.



On prend m , milieu de $[a, b]$ et on calcule $f(m)$



Ici la fonction s'annule entre a et m .

Il suffit donc de réitérer le processus en prenant comme nouvelle valeur de b la valeur de m .

On continue ainsi de suite tant que $b - a > eps$.

Un programme implémentant la méthode de dichotomie a, typiquement, la structure suivante :

```
1 def fTest(x):
2     return x*x - 2
3
4 def dichotomie(f, a, b, eps) :
5     while b - a > eps :
6         m = (a+b)/2 # milieu de [a,b]
7         if f(m)==0 :
8             return m
9         elif f(a)*f(m)< 0 :
10            b = m
11        else:
12            a = m
13    return (a+b)/2
14
15 solution = dichotomie(fTest, 0, 4, 0.001)
```

```
16 print(solution)
```

1.2 Méthode de Newton

Je vous la présente quand même... ceci étant, dans l'étude de systèmes chimiques, vous utiliserez exclusivement la méthode de dichotomie¹.

On cherche à trouver une bonne approximation du zéro d'une fonction d'une variable réelle $f(x)$ en considérant son développement de TAYLOR au premier ordre. Pour cela, partant d'un point x_0 , on approche la fonction au premier ordre : $f(x) \approx f(x_0) + f'(x_0)(x - x_0)$.

Pour trouver un zéro de cette fonction d'approximation, il suffit de calculer l'intersection de la droite tangente avec l'axe des abscisses.

Il suffit donc de résoudre l'équation :

$$0 = f(x_0) + f'(x_0) \times (x - x_0).$$

On trouve ainsi un point (si la dérivée en

x_0 n'est pas nulle!) $x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$. Cette

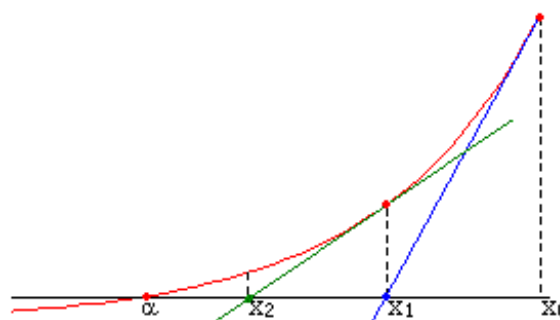
relation traduit la définition même de la dérivée :

$$f'(x_0) = \frac{f(x_0) - 0}{x_0 - x_1}.$$

On réitère le processus afin de construire, par

récence, une suite : $x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$.

On arrête lorsque la valeur absolue de la différence entre x_{k+1} et x_k est inférieure à la précision souhaitée sur la détermination du zéro.



Un programme implémentant la méthode de NEWTON a, typiquement, la structure suivante :

```
1 def fTest(x):
2     return x*2 - 2
3
4 def deriveTest(x):
5     return 2*x
6
7 def newton(f,df,x0,deltaX):
8     x1=x0-f(x0)/df(x0)
9     while abs(x1-x0)>deltaX:
10         x0=x1
11         x1=x0-f(x0)/df(x0)
12     return x1
13
14 solution = newton(fTest, deriveTest, 2, 1e-8)
15 print(solution)
```

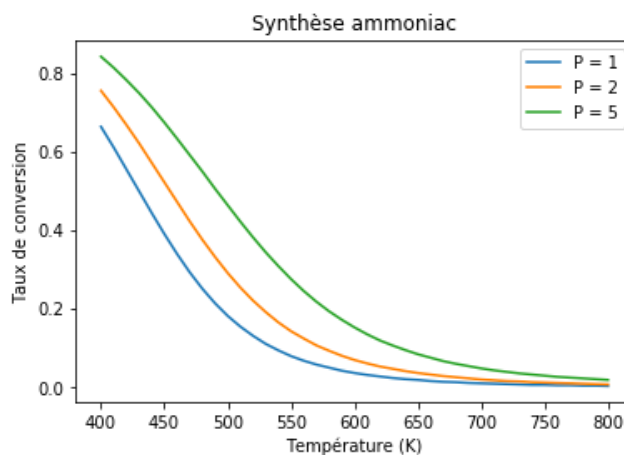
1. Pour des applications « chimiques », je n'ai vu d'utilisation de la méthode de NEWTON que pour l'étude du zéro d'une fonction de plusieurs variables. Pour le coup, la méthode de dichotomie ne peut pas être exploitée efficacement

2 Trois challenges de chimistes. . .

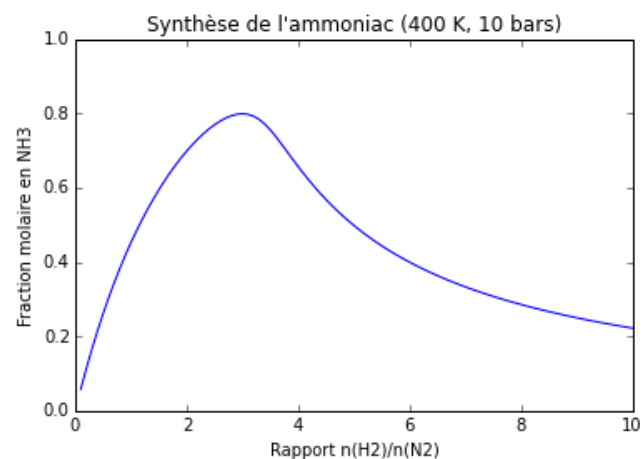
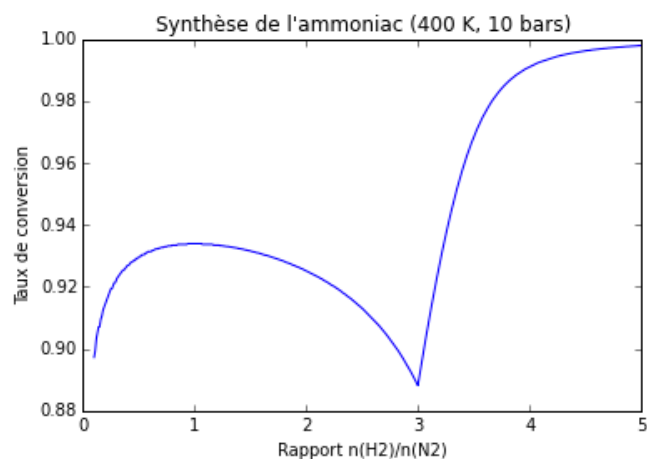
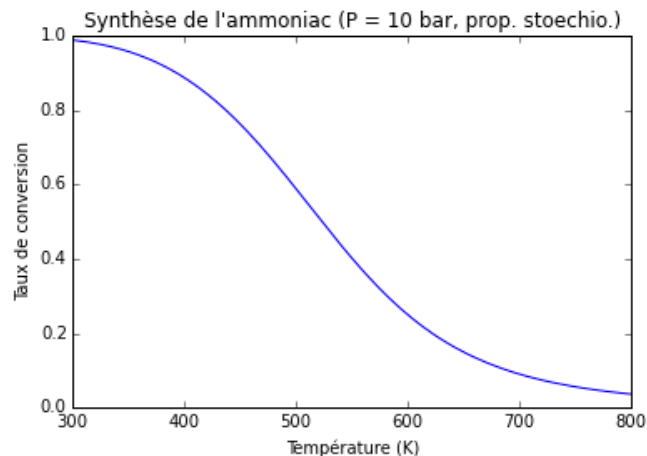
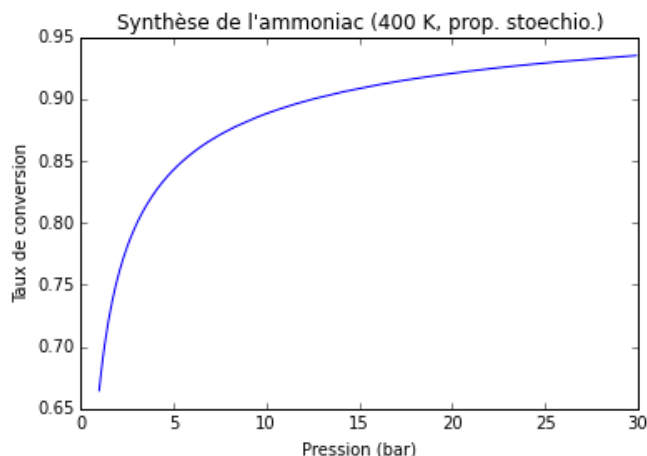
2.1 Taux de conversion en fonction de la température

On considère l'équilibre de synthèse de l'ammoniac pour lequel $\Delta_r H^0 = -92 \text{ kJ.mol}^{-1}$ et $\Delta_r S^0 = -200 \text{ J.K}^{-1}.\text{mol}^{-1}$.

On part des proportions stoechiométriques.
On étudie le taux de conversion en fonction pour différentes valeurs de la pression totale.
On désire obtenir la courbe ci-contre.
A vous de jouer !



Après on peut tout faire pour son cours sur les déplacements d'équilibre !



2.2 A la recherche de l'hétéroazéotrope

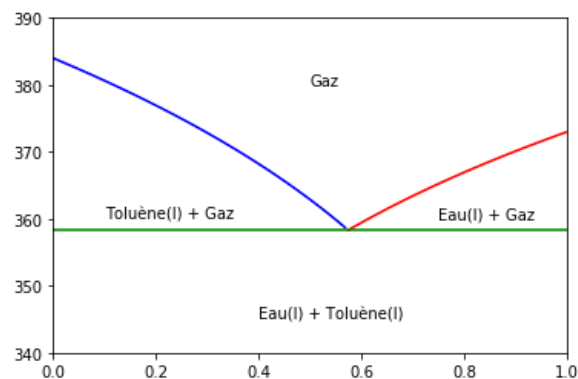
Soit par exemple le mélange binaire eau-toluène.

On donne l'enthalpie standard de vaporisation et la température d'ébullition :

- pour le toluène : $LA = 38 \text{ kJ.mol}^{-1}$, $TA = 384 \text{ K}$
- pour l'eau : $LB = 42 \text{ kJ.mol}^{-1}$, $TB = 373 \text{ K}$

L'objectif est de tracer le diagramme binaire toluène-eau.

L'instruction `plt.annotate('Gaz', (0.5, 380))` permet d'écrire le libellé souhaité aux coordonnées souhaitées.



2.3 Courbe de titrage

Ecrire un programme simulant le titrage d'un monoacide faible par une base forte.

Pour experts... en exploitant les résultats de l'activité 5 sur les courbes de répartition, on peut envisager le titrage d'un polyacide quelconque ; voire même le titrage d'un mélange d'acides !

Si on arrive là, on peut « sans trop » de difficultés envisager le titrage de n'importe quoi par n'importe quoi pour peu qu'une seule particule soit échangée. Chiche !

3 Quelques pistes

3.1 Synthèse de l'ammoniac

Si on ne souhaite tracer que le premier graphe, on peut faire un peu ce qu'on veut. En anticipant sur le suivant et un peu logiquement aussi... () il faudrait :

- définir R , $DrH0$, $DrS0$ comme variables globales ;
- coder une fonction $K0(T)$;
- coder une fonction $Qr(P, ksi, n0, n1, n2)$ (où $n0$, $n1$ et $n2$ sont les quantités initiales en N_2 , H_2 et NH_3) ;
- coder une fonction $affinite(T, P, ksi, n0, n1, n2)$

3.2 A la recherche de l'hétéroazéotrope

Après avoir défini une fonction $osee(x, Teb, Lvap)$, pour trouver l'intersection des courbes de rosée, toute l'astuce consiste à définir une fonction auxiliaire retournant la différence entre la courbe de rosée relative à l'eau et celle relative au toluène.

3.3 Courbe de titrage

L'idée est de résoudre l'équation d'électroneutralité par dichotomie.

- on écrit l'équation d'électroneutralité ;
- on écrit l'équation de conservation sur le couple acide/base étudié ce qui permet d'obtenir une équation dont l'unique inconnue est h ;
- on écrit une fonction de dichotomie (celle-ci doit travailler sur le pH et non sur h !) ;
- on boucle sur les différentes valeurs du volume.

4 Solutions

4.1 Synthèse de l'ammoniac

```

1  """
2  dichotomie
3  """
4  def dichotomie(f, xmin, xmax, delta):
5      while xmax - xmin > delta:
6          if f(xmin)*f((xmin+xmax)/2) < 0:
7              xmax = (xmin+xmax)/2
8          else:
9              xmin = (xmin+xmax)/2
10         return (xmin+xmax)/2
11
12     """
13     diagramme binaire
14     """
15     R = 8.314
16     TA = 384      #toluène
17     LA = 38000
18     TB = 373      #eau
19     LB = 42000

```

```

20
21 def rosee(x,Teb, Lvap):
22     return Teb/(1 - (R*Teb*log(x)/Lvap))
23
24 def fonction(x):
25     return rosee(x, TB, LB)-rosee(1-x, TA, LA) # attention au
        1-x !!!
26
27 XH = dichotomie(fonction, 0.1, 0.9, 0.001)
28 TH = rosee(XH,TB,LB)
29
30 print('Hétéroazéotrope : xH =', XH, ' TH = ', TH)
31
32 Xeau = [XH + i *(1 - XH)/100 for i in range(101) ]
33 Roseau = [rosee(x,TB,LB) for x in Xeau ]
34 Xtol = [i *XH/100 for i in range(100) ]
35 Rostol = [rosee(1-x,TA,LA) for x in Xtol ]
36
37 plt.plot(Xeau, Roseau, 'r')
38 plt.plot(Xtol, Rostol, 'b')
39 plt.plot([0,1], [TH,TH], 'g')
40 plt.axis([0, 1, 340, 390])
41 plt.annotate('Gaz', (0.5, 380) )
42 plt.annotate('Eau(1) + Toluène(1)', (0.4, 345) )
43 plt.annotate('Eau(1) + Gaz', (0.75, 360) )
44 plt.annotate('Toluène(1) + Gaz', (0.1, 360) )
45 plt.show()

```

4.2

A la recherche de l'hétéroazéotrope

```

1  #variables globales utilisées dans l'ensemble du programme
2  R = 8.314
3  n1 = 1.0 # quantité initiale en N2
4  n2 = 3.0 # quantité initiale en H2
5  n3 = 0.0 # quantité initiale en NH3
6  Tmin = 400.0 #température minimale de l'étude
7  Tmax = 800.0 #température maximale de l'étude
8  deltaT = 10.0 #intervalle de température entre deux calculs
9
10 DrH0 = - 92000 # enthalpie standard de réaction en J/mol
11 DrS0 = - 200 # entropie standard de réaction en J/K/mol
12
13 def A(ksi, T, P):#calcul de l'affinité chimique de la réaction
        en fonction de l'avancement, de T, de P
14     return -(DrH0 - T*DrS0)
        -R*T*log(pow(n3+2*ksi,2)*pow(n1+n2+n3-2.0*ksi,2.0)/((n1-ksi)*pow(n2
15
16 def ksi_eq(mini, maxi, precision, T, P):#calcul de
        l'avancement à l'équilibre entre mini et maxi avec la
        precision souhaitée
17     while maxi - mini > precision:
18         milieu = (maxi + mini)/2
19         if A(mini, T, P)*A(milieu, T, P) <0 :

```

```

20         maxi = milieu
21     else:
22         mini = milieu
23     return (maxi + mini)/2
24
25 def plot_to(P):#abscisses et ordonnées d'une courbe taux de
    conversion = f(T) pour P donnée
26     X=[]
27     Y=[]
28     T= Tmin
29     while T <= Tmax :
30         to = ksi_eq(0.00001, min(n1, n2/3), 0.001, T, P)
31         X.append(T)
32         Y.append(to)
33         T = T + deltaT
34     return X, Y
35
36
37 for P in [1, 2, 5]:# on trace la courbe pour les différentes
    valeurs de la pression (en bar)
38     X, Y = plot_to(P)# on récupère abscisses et ordonnées
39     plt.plot(X, Y, label = "P = "+str(P))# et on les traces
40 plt.xlabel('Température (K)') ;
41 plt.ylabel('Taux de conversion') ;
42 plt.legend()
43 plt.title('Synthèse ammoniac') ;
44 plt.show()

```

4.3 Courbe de titrage

```

1 Ke = pow(10,-14)
2
3 """ routines utilitaires pour calculer le pH d'un mélange """
4 def f(h,C0,V0,Ct,Vt,Ka): # équation à annuler pour déterminer
    le pH par dichotomie
5     return (h - Ke/h)*(V0 + Vt) + Ct*Vt - C0*V0/(1 + h/Ka)
6
7 def pH(C0,V0,Ct,Vt,Ka): # méthode de dichotomie pour calculer
    le pH
8     a=0
9     b=14
10    fa = f(pow(10,-a), C0, V0, Ct, Vt, Ka)
11    while b - a > 0.01 :
12        c = (a+b)/2
13        fc = f(pow(10,-c), C0, V0, Ct, Vt, Ka)
14        if (fa *fc < 0):
15            b = c
16        else:
17            a = c
18            fa = fc
19    return (a+b)/2
20
21 """ programme principal """

```

```
22 V = [] # tableau de valeur pour stocker les volumes
23 PH = [] # tableau de valeur pour stocker les pH correspondants
24 v = 0 # volume "courant"
25 Ca = 0.01 #concentration en acide titré
26 Va = 100 #volume titré
27 Cb = 0.1 #concentration en base
28 pas = 0.01 #pas d'incrémentement du volume
29 Vmax = 25 # volume maximal de réactif titrant
30
31 while v <= Vmax :
32     V.append(v)
33     ppH = pH(0.01, 100, 0.1, v, pow(10,-4))
34     PH.append(ppH)
35     v = v + pas
36
37 plt.plot(V, PH)
38 plt.show()
```