

## 3 Acides faibles dans l'eau

### Programmes à compléter

L'objectif de cette activité est de déterminer le taux d'avancement final de la réaction entre un acide faible et l'eau puis de tracer le diagramme de distribution du couple acide-base dans l'eau.

#### Fichiers Python

Programmes à compléter  
Fiches d'accompagnement

[hatier-clic.fr/pct210](http://hatier-clic.fr/pct210)

#### Prérequis théoriques

- Constante d'acidité d'un couple acide-base
- pH
- Taux d'avancement final
- Équations du deuxième degré

On considère l'acide faible d'un couple de constante d'acidité  $K_A$  (donnée par la valeur du  $pK_A$ ). La concentration apportée en acide faible est notée  $c_a$ .

La question **1c** consiste à établir l'équation du deuxième degré dont une solution est la concentration en ions oxonium une fois atteint l'équilibre de la réaction entre l'acide et l'eau. Cette concentration est notée  $h$ .

La question **1e** conduit à compléter le programme `tauxavancementfinal.py` pour résoudre cette équation. Le programme calcule le taux d'avancement final et le pH à l'équilibre de la solution.

À la question **2**, le programme `diagrammedistribution.py` permet de tracer le diagramme de distribution de ce couple acide-base dans l'eau.

Les questions **2a** à **2e** permettent de comprendre quels calculs sont faits pour tracer ce diagramme.

## Programme à compléter – *taux d'avancement final.py*

### Module importé

Le module `math` fournit les instructions mathématiques utiles à l'activité, telles que la fonction racine carrée `sqrt()` et la fonction logarithme décimal `log10()`.

### Entrée des données et calcul de $K_A$

La fonction `input()` permet à l'utilisateur de renseigner manuellement la valeur du  $pK_A$  et la concentration apportée  $c_a$  (notée concentration dans le programme). La fonction `float()` permet d'assurer que les valeurs entrées seront bien considérées comme des nombres à virgule. La constante d'acidité  $K_A$  est ensuite calculée.

### À modifier : résolution de l'équation du deuxième degré

Les lignes 17 à 19 sont à compléter à la question 1c avec les expressions des coefficients du polynôme du deuxième degré déterminées à la question 1b. Les lignes 21, 23 et 25 sont à compléter avec les expressions permettant de calculer le discriminant, la valeur de la concentration en ion oxonium et le taux d'avancement final. Le pH est ensuite calculé.

### Ne pas modifier : affichage des résultats

Le taux d'avancement final (en pourcentage), le pH et les concentrations finales de l'acide et de la base sont affichés avec deux chiffres après la virgule :

- la fonction `round(X, n)` arrondit X à n chiffres après la virgule ;
- la fonction `format(X, "n.mE")` présente X en notation scientifique, avec n chiffres en tout, dont m après la virgule.

```

1  from math import *
2
3  print("")
4  print("*****")
5  print("** Taux d'avancement final **")
6  print("*****Hatier 2020**")
7
8  print("")
9  print("Attention : le separateur decimal est le point")
10 print("")
11
12 pKa=float(input("pKa du couple : "))
13 concentration=float(input("Concentration apportee en mol/L : "))
14 Ka=10**(-pKa)
15 ### A MODIFIER ###
16 ### Coefficients du polynome du deuxieme degre
17 a=...
18 b=...
19 c=...
20 ### Discriminant
21 Delta=...
22 ### Solution pour [H3O+]
23 h=...
24 ### Taux d'avancement final
25 tau=...
26 ### pH final
27 pH=-log10(h)
28
29 ### NE PAS MODIFIER ###
30
31 ### Ecriture des resultats
32 print("")
33 print("Taux d'avancement final :",round(tau*100,2),"%")
34 print("pH final :",round(pH,2))
35 print("Concentration finale de l'acide :",format(concentration-h,"3.2E"),"mol/L")
36 print("Concentration finale de la base :",format(h,"3.2E"),"mol/L")

```

## Programme – *diagrammedistribution.py*

### Module importé

Le module `pylab` fournit les instructions graphiques utiles à l'activité.

### Entrée de la donnée du $pK_a$

La fonction `input()` permet à l'utilisateur de renseigner manuellement la valeur du  $pK_a$ . La fonction `float()` permet d'assurer que les valeurs entrées seront bien considérées comme des nombres à virgule.

**Attention** le séparateur décimal est le point, pas la virgule.

### Initialisation

Les trois listes utilisées (`pH`, pourcentages `Pa` et `Pb`) sont initialisées.

### Calcul des pourcentages

Les questions **2b** à **2e** permettent de comprendre ces expressions des pourcentages de l'espèce acide conjuguée  $P_a$  et de l'espèce basique conjuguée  $P_b$  calculées ici.

### Tracés

La fonction `plot()` permet le tracé des courbes d'évolution des pourcentage des espèces acide et basique en fonction du  $pH$  dans le diagramme de distribution.

```

1  from pylab import *
2
3  print("")
4  print("*****")
5  print("* Diagramme de distribution *")
6  print("*****Hatier 2020**")
7
8  print("")
9  print("Attention : le separateur decimal est le point")
10 print("")
11
12 pKa=float(input("pKa du couple : "))
13
14 Pb=[]
15 Pa=[]
16 pH=[]
17
18 for i in range(120):
19     pH.append(1+i/10)
20     Pa.append(100/(1+10**(pH[i]-pKa)))
21     Pb.append(100-Pa[i])
22
23 ### Ecriture des resultats
24 print("")
25
26 xlim(0,14)
27 ylim(-1,101)
28 plot(pH,Pa,"r",label="Espèce acide")
29 plot(pH,Pb,"b",label="Espèce basique")
30 xlabel("pH")
31 ylabel("Pourcentages")
32 legend()
33 grid(True)
34 show()
```