

## 46 Simuler une courbe de titrage conductimétrique

### Programme à compléter

L'objectif de cet exercice est de tracer une courbe de titrage conductimétrique simulée par un programme Python.

#### Fichiers Python

Programme à compléter  
Fiche d'accompagnement  
[hatier-clic.fr/pct108](http://hatier-clic.fr/pct108)

#### Prérequis théoriques

- Principe d'un titrage
- Loi de Kohlrausch

La question **a** implique d'obtenir les valeurs des quantités de matière des quatre ions (sodium, hydroxyde, sulfate, fer II) impliqués dans ce titrage, pour chaque volume de solution titrante versé. Les expressions sont différentes avant et après l'équivalence.

On obtient ainsi la courbe de titrage simulée en supposant la loi de Kohlrausch valide.

On peut alors faire varier le volume d'eau additionnel pour constater son influence sur la courbe.

### Programme à compléter

```

1  from pylab import *
2
3  print("")
4  print("*****")
5  print("* Quantites de matiere et titrage *")
6  print("* Exercice 46 p. 108 *")
7  print("* Equation de la reaction : *")
8  print("* Fe2+ + 2HO- --> Fe(OH)2 *")
9  print("*****Hatier 2020**")
10
11 print("")
12 print("Attention : le separateur decimal est le point")
13 print("")
14
15 ### A MODIFIER : DONNEES ###
16 ### Concentration de la solution titree en mol/L
17 c1=0.2
18 ### Volume de solution titree en mL
19 V1=20.0
20 ### Concentration de la solution titrante en mol/L
21 c=1.0
22 ### Volume maximal affiche en mL
23 Vmax=25.0
24 ### Volume d'eau ajoute en mL
25 Veau=150
26
27 ### NE PAS MODIFIER : CALCULS ###
28 ### Calcul de la quantite de matiere apportee de reactif titre en mol
29 n1=c1*V1/1000
30 ### Calcul du volume equivalent en mL
31 Ve=2*c1*V1/c

```

#### Module importé

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

#### Données (à modifier si nécessaire)

L'énoncé n'amène pas spécialement à modifier les données, hormis le volume d'eau ajouté.

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

Prendre garde aussi au fait que les volumes sont en millilitres, pas en litres.

#### Ne pas modifier : calculs

L'objet de cet exercice n'est pas de modifier le calcul du volume équivalent et de la quantité de matière apportée de réactif titré  $n_1$  (voir l'exercice 45 p. 107).

**Ne pas modifier : initialisation des listes**

Le volume et les quatre quantités de matière vont varier. Ces grandeurs sont construites sous forme de listes.

**À compléter : calculs des quantités de matière avant l'équivalence**

Avant l'équivalence l'avancement maximal  $x_{\max}$  de la réaction de titrage varie. Il faut donc le calculer à chaque volume de solution titrante apporté et exprimer les quantités de matière des espèces ioniques en fonction de cet avancement maximal.

La boucle `while` incrémente du pas en volume  $dV$  la valeur du dernier volume présent dans la liste et complète les listes des quantités de matière, tant que le volume est inférieur au volume équivalent.

**À compléter : calculs des quantités de matière après l'équivalence**

Avant l'équivalence l'avancement maximal  $x_{\max}$  de la réaction de titrage est constant. Il est en l'occurrence égal à la  $n_1$ , quantité de matière de réactif titré apportée. Toutes les quantités de matière s'expriment donc en fonction de  $n_1$  (ou de  $V_E$  volume équivalent).

La boucle `while` incrémente du pas en volume  $dV$  la valeur du dernier volume présent dans la liste et complète les listes des quantités de matière, tant que le volume est inférieur au volume maximal affiché (volume total de la burette).

**Ne pas modifier : calcul du volume total et de la conductivité**

La boucle `for` crée, pour chaque volume de solution titrante ajouté, un élément dans la liste  $V_{\text{tot}}$ , volume total, en mètres cube, et dans la liste  $\sigma$ , valeur de la conductivité en siemens par mètre.

Les dernières lignes tracent le graphique.

```

33  ### NE PAS MODIFIER : INITIALISATION
34  ### Volume de solution titrante verse
35  V=[0]
36  ### Quantite de matiere des especes ioniques en mol
37  nsulfate=[n1]
38  nfer=[n1]
39  nsodium=[0]
40  nhydroxyde=[0]
41  ### Pas en volume en mL
42  dV=0.5
43  ### Increment
44  i=0
45
46  ### A MODIFIER : BOUCLE
47  ### Rappels : L[-1] renvoie le dernier element de la liste L
48  ### et L.append(X) ajoute X a la fin de la liste L
49  while V[-1]<Ve:
50      i=i+1
51      V.append(V[-1]+dV)
52      xmax=...
53      nfer.append(...)
54      nsulfate.append(...)
55      nsodium.append(...)
56      nhydroxyde.append(...)
57
58  while V[-1]<Vmax:
59      i=i+1
60      V.append(V[-1]+dV)
61      nfer.append(...)
62      nsulfate.append(...)
63      nsodium.append(...)
64      nhydroxyde.append(...)
65
66  ### NE PAS MODIFIER ###
67
68  ### Ecriture des resultats
69  print("")
70  print("Volume equivalent :",round(Ve,2),"mL")
71  xlim(0,Vmax)
72
73  ### Conductivites molaires ioniques en S.m2/mol
74  lsulfate=16.0e-3
75  lfer=10.8e-3
76  lsodium=5.0e-3
77  lhydroxyde=19.8e-3
78
79  ### Initialisations Vtot en m3, sigma en S/m
80  Vtot=[]
81  sigma=[]
82
83  for i in range(len(V)):
84      Vtot.append((V[i]+Vl+Veau)/1e6)
85      sigma.append((lsulfate*nsulfate[i]+lfer*nfer[i]+lsodium*nsodium[i]
86                  +lhydroxyde*nhydroxyde[i])/Vtot[i])
87
88  xlabel("V (en mL)")
89  ylabel("σ (en S/m)")
90  plot(V,sigma)
91  show()

```