

Appréhender l'équivalence d'un titrage

Capacité numérique : Représenter, à l'aide d'un langage de programmation, l'évolution des quantités de matière des espèces en fonction du volume de solution titrante versé.

Le but de cette activité est de tracer l'évolution des quantités de matière des espèces au cours d'un titrage pour comprendre la notion d'équivalence.

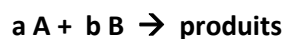
On s'intéresse au titrage d'un réactif titrant A par un réactif titré B.

Soit V1 le volume de réactif titré initial.

Soit C2 la concentration de réactif titrant.

Soit Vbe le volume de réactif titrant versé à l'équivalence.

L'équation modélisant la réaction servant de support au titrage se trouve ci-dessous.



1. Rappeler la définition de l'équivalence.
2. Ouvrir le programme : « **titrage_eleve.py** »
3. Compléter le code permettant le calcul de C1 la concentration de réactif titré en utilisant la bonne formule.

On pose :

Nb_stoech1 = a

Nb_stoech2 = b

```
#-----  
# Calcul de la concentration de réactif titré  
# A compléter  
#-----  
  
print("La concentration de réactif titré est : ", C1, " mol/L.")  
print()
```

4. Boucle simulant l'ajout d'un volume V2 de solution titrante :

Pour chaque volume V2 de solution titrante versé, le programme va devoir calculer les quantités de matière de réactifs finales et tracer une courbe représentant ces quantités en fonction de V2 versé.

On suppose que l'on verse le volume de solution titrante mL par mL jusqu'au volume maximal de la burette c'est-à-dire 25 mL.

Travail à réaliser niveau **EXPERT** :

Dans la partie « Boucle simulant l'ajout d'un volume V2 de solution titrante » écrire un programme qui, pour chaque volume V2 versé, calcule N1_restant et N2_restant. Ces valeurs sont ensuite mises dans deux listes N1 et N2 avec la méthode append.

Placer des croix sur une courbe aux coordonnées (V2_verse, N1) et (V2_verse, N2) (en rouge pour N1 et bleu pour N2).

Il faudra penser qu'une quantité finale de matière ne peut pas être négative (dans ce cas, elle est nulle).

Travail à réaliser niveau **AVANCÉ** :

L'algorithme, en langage naturel, permettant de répondre à l'objectif est le suivant :

Pour chaque volume V_2 compris entre 0 et $V_{2_max} + 1$:

Calculer $N1_ini$ et $N2_verse$ à partir des volumes et concentrations

Calculer $N1_restant$ et $N2_restant$

Ces valeurs sont ensuite mises dans deux listes $N1$ et $N2$ avec la méthode `append`.

Il faudra penser qu'une quantité finale de matière ne peut pas être négative (dans ce cas, elle est nulle).

Traduire cet algorithme en Python et taper le code dans la partie « Boucle simulant l'ajout d'un volume V_2 de solution titrante » à compléter.

Travail à réaliser niveau **DÉBUTANT** :

Dans la partie « Boucle simulant l'ajout d'un volume V_2 de solution titrante » à compléter.

- Taper le code pour créer une boucle pour V_2 allant de 0 à $V_{2max} + 1$.

- Dans cette boucle : créer une variable $N1_init$ qui vaut $C1 \cdot V1 / 1000$. Faire la même chose pour $N2_verse$

Calculer $N1_restant$ et $N2_restant$ si $V_2 < V_{be}$ et si $V_2 > V_{be}$

Il faudra penser qu'une quantité finale de matière ne peut pas être négative (dans ce cas, elle est nulle).

Ces valeurs sont ensuite mises dans deux listes $N1$ et $N2$ avec la méthode `append`.

Questions : Analyse du graphe obtenu

1 . PARTIE GAUCHE DE LA COURBE :

a. Justifier le fait que la quantité de réactif à titrer diminue.

b. Quel est le réactif limitant sur cette partie ?

2. PARTIE DROITE DE LA COURBE :

a. Justifier le fait que la quantité de réactif titrant augmente.

b. Quel est le réactif limitant sur cette partie ?

3. Pour quel volume de solution titrante versé les réactifs ont-ils été introduits dans les proportions stœchiométriques ?

AIDE PYTHON

LECTURE DES DONNEES

```
variable = input("texte")
```

AFFICHAGE

```
print("texte", variable)
```

Conditions

```
if A>B:
    Instructions à
    réaliser si A est
    supérieur à B
else:
    Instructions à
    réaliser si ce
    n'est pas le cas
```

Créer une boucle bornée

```
for i in range(début, fin):
```

Taper ensuite les instructions à réaliser pendant la boucle.
Le compteur "**i**" va prendre des valeurs entières entre "**début**" et "**fin-1**".
Bien penser à indenter.

Placer une croix au point de coordonnée X,Y

```
plt.plot(X,Y,"kx")
```

"kx" signifie que les points affichés seront noirs ("k") et représentés par des croix ("x")

Autres options possibles :

Tracé		Type de points tracés					Couleurs				
-	--	o	.	x	+	v	r	b	g	k	m
Points reliés	Points reliés en pointillé	Gros « ronds »	Petit point	Croix	Croix +	Triangle	Rouge	Bleu	vert	noir	magenta

L.append(x)

Ajoute l'élément x en dernière position de la liste L.