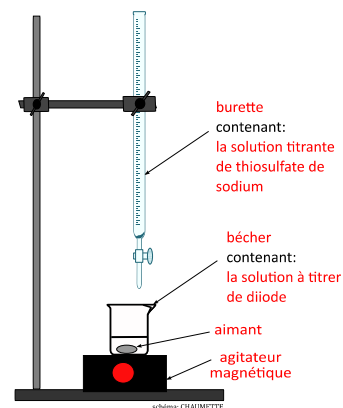


Appréhender l'équivalence d'un titrage

Niveau (Thèmes)	1 ^{ère} spécialité (Suivi de l'évolution d'un système chimique) Ou Terminale spécialité (Analyser un système par des méthodes chimiques)
Introduction	Lors d'un titrage, la détermination de la composition du milieu réactionnel, pour chaque volume V2 de solution titrante versée, permet de bien comprendre le passage à l'équivalence. Il est impossible de réaliser « à la main », les calculs de quantités de matière pour chaque volume V2 versé. L'utilisation d'un programme permet, à l'aide d'une boucle, d'automatiser ces calculs.
Type d'activité	Activité de programmation en salle informatique
Compétences	ANALYSER : - Proposer les étapes d'une résolution - Repérer ou sélectionner des informations utiles RÉALISER : - faire un calcul littéral et un calcul numérique - Utiliser une relation entre grandeurs physiques / Etablir une relation littérale - Faire une détermination graphique
CRCN - Compétences Num.	3. Création de contenus 3.4. Programmer
Notions et contenus du programme	Première : Déterminer la composition de l'état final d'un système siège d'une transformation chimique totale à l'aide d'un langage de programmation. Relier qualitativement l'évolution des quantités de matière de réactifs et de produits à l'état final au volume de solution titrante ajoutée. Relier l'équivalence au changement de réactif limitant et à l'introduction des réactifs en proportions stœchiométriques. Établir la relation entre les quantités de matière de réactifs introduites pour atteindre l'équivalence. Terminale : Utiliser un langage de programmation pour représenter l'évolution des quantités de matière des espèces intervenant dans un titrage en fonction du volume de solution titrante versée. Le tracé de cette évolution doit amener l'élève à comprendre ce que représente l'équivalence et déterminer, pour un volume de solution titrante ajouté, la composition du milieu réactionnel.
Objectif(s) pédagogique(s)	La boucle au cœur du programme permet de comprendre qu'un titrage peut être vu comme une succession de réactions avec des quantités de matières initiales différentes.
Objectifs disciplinaires	Découvrir l'équivalence d'un titrage grâce à la programmation
Description succincte de l'activité	L'activité demande, dans un premier temps, de programmer la recherche de réactif limitant. Si les élèves sont en terminale, cette partie a déjà été traitée en 1 ^{ère} spé (capacité numérique exigible). Elle permet aux élèves de bien maîtriser la méthode de détermination de x_{\max} . Puis l'activité demande à réutiliser ce programme au sein d'une boucle qui simule l'ajout d'un certain volume de solution titrante. Le tracé des quantités de matière en fonction du volume versé permet de dégager la notion d'équivalence. A la fin de la séance, le programme choisit au hasard une concentration pour que l'élève la détermine à l'aide des courbes qu'il a tracées.
Découpage temporel de la séquence	Durée Prévue : 1h10 – Différenciation prévue - Ecriture de l'algorithme en langage « naturel » : 15 min - Programmation de l'algorithme en Python : 15 min - Codage de la boucle : 20 min - Interprétation : 10 min - Calcul de C1 inconnu à partir d'un exemple « automatique » : 10 min
Pré-requis	<i>Physique-chimie</i> : Quantités de matière, Avancement, avancement maximal. <i>Python</i> : notion de variables, conditions, boucles bornées. <i>Fourni aux élèves</i> : un mini-mémento permettant de tracer une courbe avec Matplotlib.
Outils numériques utilisés/Matériel	Edupython et ordinateur sous Windows
Gestion du groupe Durée estimée	- Travail en binôme en présentiel et en autonomie. Différenciation incluse. - Durée 1h10

Énoncé à destination des élèves

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.



1. Ecriture d'un code python permettant de déterminer un réactif limitant

1.1. Travail préliminaire

On s'intéresse au titrage du diiode I_2 par les ions thiosulfate $S_2O_3^{2-}$. L'équation modélisant la réaction servant de support au titrage se trouve dans le tableau d'avancement ci-dessous.

1. Compléter, *de manière littérale*, le tableau d'avancement suivant :

	I_2	+	$2 S_2O_3^{2-}$	\rightarrow	$2 I^-$	+	$S_4O_6^{2-}$
Etat initial	n1_ini		n2_ini				
Etat intermédiaire							
Etat final	n1_fin =		n2_fin =		n3_fin =		n4_fin =

2. Écrire, *en langage naturel*, un algorithme permettant de déterminer x_{max} connaissant $n1_ini$ et $n2_ini$

3. Généraliser votre algorithme en introduisant des nombres stœchiométriques quelconques : **sto1** et **sto2** (n'écrire que les parties modifiées)

1.2. Programme Python

Ouvrir Edupython et commencer à saisir le code ci-contre et le sauvegarder sous le nom « **reactif_limitant_titrage.py** » :

4. Compléter le code en traduisant, en Python, votre algorithme du paragraphe précédent.

5. Compléter le code en faisant calculer $n1_fin$ et $n2_fin$.

6. Faire afficher x_{max} , $n1_fin$ et $n2_fin$. A ce stade, le programme doit afficher :

```
# Initialisation des variables :
sto1=1
sto2=2
n1_ini=2
n2_ini=2
xmax=0

# Calcul de xmax, n1_fin et n2_fin:
```

```
xmax= 1.0 mol
n1_fin= 1.0 mol
n2_fin= 0.0 mol
>>>
```

2. Code permettant de calculer les quantités de matière au cours d'un titrage

2.1. Protocole du titrage qui sera à simuler sous Python

- Prélever avec une pipette jaugée $V_1 = 5$ mL de solution de diiode de concentration $C_1 = 0,01$ mol.L⁻¹
- Verser ces 5 mL dans un bécher et ajouter environ 20 mL d'eau.
- Remplir la burette avec une solution de thiosulfate de sodium de concentration $C_2 = 0,01$ mol.L⁻¹.
- Verser délicatement le thiosulfate de sodium, mL par mL, jusqu'au volume maximal de la burette : $V_{2max} = 25$ mL

7. Charger le code : « **Chimie_Evol_n_Titrage.py** »

8. Compléter la partie «initialisation de données» à partir des données du protocole de titrage (laisser les volumes en mL).

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

9. Objectif :

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.

Travail à réaliser niveau **EXPERT** :

Entre les parties roses « TRAVAIL A REALISER », écrire un programme qui, pour chaque volume V2 versé, calcule $n1_{\text{final}}$ et $n2_{\text{final}}$ et place des croix sur une courbe aux coordonnées (V2, n_{final}) (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 V2 compris entre 0 et V2max+1 :

Calculer n1_ini et n2_ini à partir des volumes et concentrations

Déterminer xmax

Calculer n1_fin et n2_fin

Si n1_fin est négatif, alors c'est que n1_fin est nul. Idem pour n2_fin.

Mettre une croix rouge sur le graphe en (V2, n1_fin). Idem pour n2_fin (croix bleue)

Traduire cet algorithme en Python et taper le code entre les parties roses « TRAVAIL A REALISER ». (Pour « déterminer xmax », vous pourrez copier/coller le code du paragraphe 1.)

Travail à réaliser niveau **DÉBUTANT** :

Entre les parties roses « TRAVAIL A REALISER » :

- Taper le code pour créer une boucle pour V2 allant de 0 à V2max+1.

- Dans cette boucle : créer une variable n1_ini qui vaut $C1 \cdot V1$. Faire la même chose pour n2_ini
Copier/coller le code que vous avez programmé dans le paragraphe 1.

Taper le code pour traduire : si $n1_{\text{fin}} < 0$ alors $n1_{\text{fin}} = 0$. Idem pour n2_fin

Mettre une croix rouge sur un graphe grâce à `plt.plot(V2, n1_fin, "rx")`
idem pour n2_fin avec une croix bleue.

2.3. Légende des axes et du graphique

10. En dessous de la partie «noms des axes et titre à compléter», saisir le nom (et l'unité) des grandeurs représentées et le titre du graphique.

3. Analyse du graphe obtenu

11. PARTIE GAUCHE DE LA COURBE :

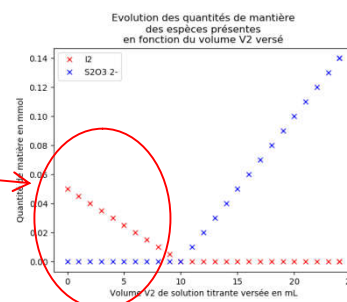
a. Justifier le fait que la quantité de réactif à titrer (I_2) diminue.

b. Qui est le réactif limitant sur cette partie ?

12. PARTIE DROITE DE LA COURBE :

a. Justifier le fait que la quantité de réactif titrant ($S_2O_3^{2-}$) augmente.

b. Qui est le réactif limitant sur cette partie ?



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

Définition : Le volume pour lequel les réactifs ont été introduits dans les proportions stoechiométriques s'appelle volume à l'équivalence.

14. Essayer de définir l'équivalence.

15. Ecrire, à l'équivalence, la relation entre les quantités de matières n_{1ini} et n_{2ini} introduites.

4. Utilisation du graphe pour déterminer la concentration inconnue d'un solution à titrer

Nous allons rendre « inconnue » la concentration C_1 . Pour cela, nous allons faire « choisir » Python une concentration au hasard.

16. Dans le code, mettre un volume V_1 de 10 mL et, à la place de la ligne $C_1=0.01$, taper :

```
import random
C1=random.randint(1,11)/1000
```

17. À la toute dernière ligne du programme (après `plt.show()`), taper : `print("C1=",C1,"mol/L")`

18. Afficher la courbe et :

- a. Déterminer le volume à l'équivalence V_{2EQ} .
- b. Écrire la relation à l'équivalence entre les quantités initiales de réactifs versés (à l'équivalence)
- c. En déduire la valeur de la concentration C_1 de la solution à titrer de I_2 .

19. Fermer la fenêtre Matplotlib : la valeur de C_1 s'affiche dans la console Python pour vérifier votre réponse.

AIDE PYTHON

AFFICHAGE

```
print("texte",variable,"autre texte",autre_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

Corrigé pour les enseignant.e.s

1. Ecriture d'un code python permettant de déterminer un réactif limitant

1.1. Travail préliminaire

1. Compléter, de manière littérale, le tableau d'avancement suivant :

	I_2	+	$2 S_2O_3^{2-}$	\rightarrow	$2 I^-$	+	$S_4O_6^{2-}$
Etat initial	n1_ini		n2_ini		0		0
Etat intermédiaire	n1_ini-x		n2_ini-2x		2x		x
Etat final	n1_fin = n1_ini-x _{max}		n2_fin = n2_ini-2x _{max}		n3_fin=2x _{max}		n4_fin=x _{max}

2. Écrire, en langage naturel, un algorithme permettant de déterminer x_{max} connaissant n1_ini et n2_ini

Il faut comparer n1_ini et n2_ini/2 :

Si n1_ini > n2_ini/2 alors :

Le réactif limitant est le 2

x_{max} = n2_ini/2

Sinon :

Le réactif limitant est le 1

x_{max} = n1_ini

Pour les élèves en difficulté :

Leur faire rédiger, avec la méthode habituelle, la recherche du réactif limitant puis leur faire traduire en condition.

3. Généraliser votre algorithme en introduisant des ombres stœchiométriques quelconques : sto1 et sto2

(n'écrire que les parties modifiées)

Dans la condition, il faut introduire les nombres stœchiométriques : « Si n1_ini/sto1 > n2_ini/sto2 alors »

Et, dans les calculs de x_{max} : Si 2 est limitant : x_{max} = n2_ini/sto2 sinon x_{max} = n1_ini/sto1

1.2. Programme Python

Ouvrir Edupython et commencer à saisir le code suivant et le sauvegarder sous le nom

« reactif_limitant_titrage » :

```
# Initialisation des variables :
sto1=1
sto2=2
n1_ini=2
n2_ini=2
xmax=0

# Calcul de xmax puis n1_fin et n2_fin :
```

```
if n1_ini/sto1 > n2_ini/sto2:
    xmax=n2_ini/sto2
else:
    xmax=n1_ini/sto1

n1_fin=n1_ini-sto1*xmax
n2_fin=n2_ini-sto2*xmax
```

4. Compléter le code en traduisant, en Python, votre algorithme du paragraphe précédent.

5. Compléter le code en faisant calculer n1_fin et n2_fin.

6. Faire afficher xmax, n1_fin et n2_fin. A ce stade, le programme doit afficher :

```
xmax= 1.0 mol
n1_fin= 1.0 mol
n2_fin= 0.0 mol
>>>
```

2. Code permettant de calculer les quantités de matière au cours d'un titrage

2.1. Protocole du titrage

7. Charger le code : « Chimie_Evol_n_Titrage.py »

8. Compléter la partie «initialisation de données» en analysant le protocole de titrage (laisser les volumes en mL).

```
""" init des données"""
sto1=1
sto2=2
C1=0.01
V1=5 # V1 laissé en mL
C2=0.01
V2max=25 # en mL
reactif1="I2"
reactif2="S2O3 2-"
```

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

9. Objectif :

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.

```
for V2 in range(0,V2max+1):
    # Pour chaque ajout de V2,calcul des quantités de matière initiales:
    n1_ini=C1*V1
    n2_ini=C2*V2

    # PARTIE "détermination de xmax" du paragraphe 1
    if n1_ini/sto1>n2_ini/sto2:
        # réactif 2 limitant
        xmax=n2_ini/sto2
    else:
        xmax=n1_ini/sto1
        # réactif 1 limitant
    # Calcul des quantités finales
    # si n<0, c'est que n=0
    n1_fin=n1_ini-sto1*xmax
    if n1_fin<0:
        n1_fin=0
    n2_fin=n2_ini-sto2*xmax
    if n2_fin<0:
        n2_fin=0

    # affichage des points sur une courbe
    plt.plot(V2,n1_fin,"rx")
    plt.plot(V2,n2_fin,"bx")
    # FIN DE LA BOUCLE #####

plt.xlabel("Volume V2 de solution titrante
versée en mL")
plt.ylabel("Quantité de matière en mmol")
plt.title("Evolution des quantités de
mantière\n des espèces présentes \nen
fonction du volume V2 versé")
```

2.3. Légende des axes et du graphique

10. En dessous de la partie «noms des axes et titre à compléter», saisir le nom (et l'unité) des grandeurs représentées et le titre du graphique.

3. Analyse du graphe obtenu

11. PARTIE GAUCHE DE LA COURBE :

a. Justifier le fait que la quantité de réactif à titrer (I_2) diminue.

L'ajout progressif de ($S_2O_3^{2-}$) qui réagit avec I_2 fait disparaître ce dernier.

b. Qui est le réactif limitant sur cette partie ?

C'est ($S_2O_3^{2-}$) car, pour chaque V2, sa quantité de matière est nulle

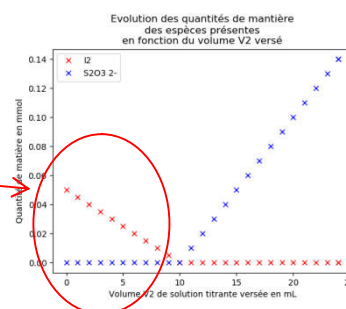
12. PARTIE DROITE DE LA COURBE :

a. Justifier le fait que la quantité de réactif titrant ($S_2O_3^{2-}$) augmente.

Il n'y a plus de I_2 dans le bécher pour réagir avec ($S_2O_3^{2-}$) donc, pour chaque ajout de V2, la qté de ($S_2O_3^{2-}$) augmente.

b. Qui est le réactif limitant sur cette partie ? C'est (I_2) car, pour chaque V2, sa quantité de matière est nulle

13. Pour quel volume de solution titrante versé les réactifs ont-ils été introduits dans les proportions stoechiométriques ? On cherche quand les 2 n_{final} sont nuls : $V_{2EQ} = 10$ mL



Définition : Le volume pour lequel les réactifs ont été introduits dans les proportions stoechiométriques s'appelle volume à l'équivalence.

14. Essayer de définir l'équivalence.

A l'équivalence, il y a changement de réactif limitant **ou bien** : à l'équivalence, les réactifs ont été introduits dans les proportions stoechiométriques

15. Ecrire, à l'équivalence, la relation entre les quantités de matières n_{1ini} et n_{2ini} introduites.

$n_{1ini} = n_{2ini}$ à l'équiv/2

4. Utilisation du graphe pour déterminer la concentration inconnue d'une solution à titrer

Nous allons rendre « inconnue » la concentration C_1 . Pour cela, nous allons faire « choisir » Python une concentration au hasard.

16. Dans le code, mettre un volume V_1 de 10 mL et, à la place de la ligne $C_1=0.01$, taper :

`C1=np.random.randint(1,11)/1000` ce code permet de tirer au sort un nombre entre 1 et 10 et de le diviser par 1000 : cela revient à prendre des C_1 au hasard entre 10^{-3} mol/L et 10^{-2} mol/L

17. À la toute dernière ligne du programme (après le `plt.show()`), taper : `print("C1=",C1)`

On tape ce code pour que les élèves puissent vérifier leur valeur de C_1 . Mais il ne s'exécute qu'une fois la fenêtre Matplotlib fermée (car placé après de `plt.show()`)

18. Afficher la courbe et :

a. Déterminer le volume à l'équivalence V_{2EQ} .

b. Écrire la relation à l'équivalence entre les quantités initiales de réactifs versés (à l'équivalence)

$n_{1ini} = n_{2ini}$ à l'équiv/2

c. En déduire la valeur de la concentration C_1 de la solution à titrer de I_2 .

$$C_1 = \frac{C_2 \times V_{2EQ}}{2V_1} \quad \text{avec } V_1 = 10 \text{ mL}$$

19. Fermer la fenêtre Matplotlib : la valeur de C_1 s'affiche dans la console Python pour vérifier votre réponse.

Pour les plus rapides, on peut imaginer une autre réaction ou leur faire recommencer plusieurs fois l'exercice.

Retour d'expérience :

Les plus-value pédagogiques (enseignants/élèves) :

Les freins :

Les leviers :

Les pistes pour aller plus loin ou généraliser la démarche :

Production d'élèves :

mettre lien, extrait de copies etc en s'assurant d'avoir les droits de diffusion auprès des élèves