**FICHE de PRÉSENTATION d’activités**

|  |  |
| --- | --- |
| ***Niveau***  | ***1ère spécialité*** |
| ***Séquence*** | ***Suivi et modélisation de l’évolution d’un système chimique*** |
| ***Titre de l’activité*** | ***Détermination de l’état final d’un système chimique à l’aide d’un programme Python*** |
| ***Type d'activité*** | ***programmation en ½ groupe*** |
| ***Références au programme*** | Notions et contenus*Avancement final, avancement maximal.**Mélanges stœchiométriques..* | Capacités exigibles*Déterminer la composition du système dans l’état final en fonction de sa composition initiale pour une transformation considérée comme totale.****Capacité numérique :*** *Déterminer la composition de l’état final d’un système siège d’une transformation chimique totale à l’aide d’un langage de programmation* |
| ***Compétences mobilisées***  | ❑ Analyser/raisonner ❑ Réaliser |
| ***Mise en œuvre*** | Pré-requis:Quelques notions de python (variable, exécution d’un programme sous Edupython)Quantités de matières |
| Durée : 1h – Différenciation : partie supplémentaire pour les plus rapides.Paragraphe 1 préparé en amont à la maisonPossibilité de faire de même pour le paragraphe 2 |
| Contraintes matérielles : salle informatique avec EduPython |
| *Liens photos* | *aucun* |
| *Auteur* | **olivier.chaumette@ac-lyon.fr** |
|  **pour le GRD groupe lycée de l’académie de LYON** |

**Fiche élève : activité**

*But de l’activité :* Pour une réaction du type a.A + b.B  c.C + d.D, écrire un algorithme permettant de déterminer le réactif limitant et le traduire en langage Python pour déterminer l’état final du milieu réactionnel.

On considère la réaction entre les ions cuivre II Cu2+ et le métal aluminium Al. Elle peut être modélisée par l’équation :

2 Als + 3 Cu2+aq  2 Al3+aq + 3 Cus

On verse **V = 100 mL** de sulfate de cuivre II de concentration molaire en soluté apporté **C = 4,0.10-1 mol/L** sur de la poudre d’aluminium de masse **mAl = 0,54 g**.

*Données :* Masses molaires : M(Al) = 27 g.mol-1 M(Cu) = 63,5 g.mol-1

 Couleur des espèces ioniques : Cu2+: bleu Al3+: incolore

**1. Travail préliminaire : (à faire à la maison)**

**1.** Déterminer les quantités de matières **initiales** de chaque réactif et produit de la réaction.

**2.** Compléter le tableau d’avancement suivant et, après avoir déterminé xmax et le réactif limitant, compléter numériquement la dernière ligne.



**2. Algorithme permettant de déterminer le réactif limitant :**

En s’appuyant sur la méthode utilisée dans la question précédente pour déterminer xmax, écrire, **en langage naturel**, un raisonnement (donc un « algorithme ») permettant de déterminer l’avancement maximal xmax et le réactif limitant dans le cas de la réaction précédente.

**3. Traduction de l’algorithme en langage Python :**

**1.** Ouvrir l’éditeur Python (Edupython) et ouvrir le programme « **1ere\_Etat\_final\_reaction\_eleves.py** »

Dans ce programme, pour chaque espèce, 4 variables sont associées. Par exemple pour le réactif 1 :

|  |  |  |
| --- | --- | --- |
| Contenu de la variable | Nom de la variable | Type |
| Formule du réactif 1 | r1 | Texte (string) |
| Quantité de matière initiale | n1\_ini | Réel (float) |
| nombre stœchiométrique | stoechio1 | Entier (int) |
| Quantité de matière finale | n1\_fin | Réel (float) |

Il en est de même pour le réactif 2, le produit 1 et le produit 2

**2.** Dans la partie «**Quantités de matière initiales**», saisir les quantités de matière initiales calculées au paragraphe 1.

 Pour saisir une puissance de 10 en python, il faut taper « e ». Exemple : 2.103 se tape : **2e3**

**3.** Dans la partie «**Nombres stoechiométriques**», saisir le nombre stœchiométrique de chaque réactif et produit.

**4.** Dans la partie «**Formules des réactifs et produits**», saisir la formule (*sans respect de mise en forme*) de chaque réactif et produit (bien les mettre en guillemets car il s’agit de textes).

**5.** Dans la partie «**Recherche du réactif limitant et de xmax**», il faut traduire l’algorithme du paragraphe 2 en langage Python (voir aide page suivante).

***Rappel de quelques correspondances entre le langage naturel et le langage Python :***

 Ne pas oublier les deux points qui indiquent le début du bloc « if »

|  |  |
| --- | --- |
| **Langage algorithmique naturel** | **Langage Python 3** |
| Si A > B alors Exécuter l’instruction 1 Exécuter l’instruction 2Sinon Exécuter l’instruction 3 Exécuter l’instruction 4Fin du si et du sinon | **if A>B :** **instruction 1**Les instructions du bloc « if » doivent être alignées l’une sous l’autre et décalées par rapport au « if » (on appelle cela **indentées**) **instruction 2****else :** **instruction 3** **instruction 4** |
| La variable « A » prend la valeur de la variable « B » | **A = B** Ne pas oublier les deux points qui indiquent le début du bloc « else »  |
| La variable entière « A » prend la valeur 2 | **A = 2** |
| La variable texte « T » prend la valeur « Bonjour » | **T = "Bonjour"** |
| La variable « n » prend la valeur « n\_ini – 2xmax» | **n = n\_ini-2\*xmax** |
| Affiche « Bonjour » | **print("Bonjour")** |
| Affiche la variable « A » en écriture scientifique à 2 chiffres significatifs | **print("%.1e"%A)** |
| Affiche la variable « B » en écriture scientifique à 3 chiffres significatifs | **print("%.2e"%B)** |

**6.** Dans la partie «**Calcul des quantités de matière finales**», suivre les consignes du programme écrites en rose : il faut écrire le code assignant aux variables de quantités de matière finales leur valeur en fonction des quantités de matières initiales, de l’avancement et des coefficients stœchiométriques.

**7.** Tester le programme. L’état final affiché doit correspondre à la dernière ligne du tableau d’avancement.

 Le concepteur du programme initial a fait le choix d’un affichage en écriture scientifique.

Ainsi, par exemple, **n(Cu)=2.0e-3.0 mol** signifie n(Cu) = 2,0.10-3 mol

**4. Utilisation du programme pour une autre réaction :**

*On s’intéresse maintenant à la réaction entre 2 moles de dioxygène gazeux et 3 moles de dihydrogène gazeux qui forme de l’eau.*

**1.** Ecrire l’équation modélisant la réaction.

**2.** Quelles sont les parties du programme Python qu’il faudra modifier.

**3.** Adapter le programme python (**comme il n’y a pas de « Produit 2 », on pourra mettre un symbole dièse # devant toutes les lignes faisant référence au Produit 2** : elles ne seront alors pas prises en compte par Python)

**4.** Le programme doit afficher comme résultat (en écriture scientifique) : xmax = 1,5 mol et H2 limitant

puis nO2 fin = 0,5 mol nH2 fin = 0 mol et nH2O fin = 3 mol

**5.** Modifier les quantités de matière initiales pour obtenir un mélange stœchiométrique.

*Remarque :* il paraît absurde d’avoir à modifier le code à chaque réaction différente. Une amélioration simple et rapide du programme consisterait à demander à l’utilisateur de saisir les quantités de matières initiales, les coefficients stœchiométriques et les formules des espèces. Ainsi, le code n’aurait plus à être modifié. Mais cela sort du cadre du programme de physique-chimie.

Pour les personnes souhaitant aller plus loin, pour demander à l’utilisateur de saisir une valeur, il suffit de taper :

**n1\_ini = float(input("Saisissez la quantité de matière initiale du réactif 1"))** pour saisir une valeur numérique et, pour saisir un texte :

**r1 = input ("Saisissiez la formule du réactif 1")**

**Fiche professeur**

On considère la réaction entre les ions cuivre II Cu2+ et le métal aluminium Al. Elle peut être modélisée par l’équation :

2 Als + 3 Cu2+aq  2 Al3+aq + 3 Cus

On verse **V = 100 mL** de sulfate de cuivre II de concentration molaire en soluté apporté **C = 4,0.10-1 mol/L** sur de la poudre d’aluminium de masse **mAl = 0,54 g**.

*Données :* Masses molaires : M(Al) = 27 g.mol-1 M(Cu) = 63,5 g.mol-1

**1. Travail préliminaire : (à faire à la maison)**

**1.** Déterminer les quantités de matières **initiales** de chaque réactif et produit de la réaction.

nCu2+ ini = CV = 0,1x4,0.10-1 = 4,0.10-2 mol nAl ini= mAl/M(Al) = 2,0.10-2 mol

**2.** Compléter le tableau d’avancement suivant et, après avoir déterminé xmax et le réactif limitant, compléter numériquement la dernière ligne.



Si Al limitant, nAl ini -2xmax = 0 donc xmax = nAl ini /2 = 0,010 mol.

Si Cu2+ limitant, nCu2+ ini -3xmax = 0 donc xmax = nCu2+ ini /3 = 0,013 mol.

Donc **Al est limitant et xmax = 0,010 mol** car, si xmax valait 0,013 mol alors nAl fin vaudrait -0,006 mol <0 : impossible !

**2. Algorithme permettant de déterminer le réactif limitant :**

Quand on analyse notre démarche, on voit qu’on compare nAl ini /2 (c’est-à-dire nbre de mole ini du réactif 1 divisé par son coef stoechiométrique) à nCu2+ini /3.

D’où l’algorithme : Et la traduction en Python :

**if n1\_ini/stoechio1 < n2\_ini/stoechio2:**

 **xmax = n1\_ini / stoechio1**

 **react\_limit = r1**

**else:**

 **xmax = n2\_ini/stoechio2**

 **react\_limit = r2**

Si nAl ini /2 < nCu2+ /3 alors

 Xmax = nAl ini /2

 Le réactif limitant est : Al

Sinon

 Xmax = nCu2+ ini /3

 Le réactif limitant est : Cu2+

**3. Traduction de l’algorithme en langage Python :**

***Les noms de variables peuvent bien sûr être modifiés pour être plus simples ou au contraire plus explicites encore.***

**Voir le code corrigé : 1ere\_Etat\_final\_reaction\_Cor\_exo1.py**

*Pour les relations à l’état final :*

nAl fin  = nAl ini – 2xmax devient : **n1\_fin =n1\_ini - stoechio1\*xmax**

nCu2+ fin  = nCu2+ ini – 3xmax devient : **n2\_fin =n2\_ini – stoechio2\*xmax**

nAl3+ fin  = nAl3+ ini + 2xmax devient : **n3\_fin =n3\_ini + stoechio3\*xmax**

nCu fin  = nCu ini + 3xmax devient : **n4\_fin =n4\_ini + stoechio4\*xmax**

*ou, bien sûr:*nAl3+ fin  = 2xmax et nCu fin  = 3xmax qui deviennent :

**n3\_fin = stoechio3\*xmax** et  **n4\_fin = stoechio4\*xmax**

*Remarque pour le professeur concernant l’affichage des résultats qui peut paraître complexe :*

L’auteur du programme a pris le parti d’afficher les résultats en écriture scientifique. C’est pourquoi il y a "%.1e"% au niveau des fonctions **print**, ce qui les rend complexes. Voici l’explication de cette instruction ainsi que le code à écrire si on ne veut pas d’écriture scientifique (**mais attention, dans ce cas, l’arrondi de Python peut être étonnant**…)

**"%.1e"%** suivi d’une variable écrit la variable en écriture scientifique avec 1 chiffre après la virgule (donc 2 chiffres significatifs au total : 2 CS). Pour avoir 3 chiffres significatifs, il faut taper "%.2e"% suivi du nom de la variable.

*Exemples :*

**A=2000 B = 0.001898**

**print("%.1e"%A) print("%.2e"%B)**

donne 2 CS en écriture scientifique :2.0e3.0 donne 3 CS en écriture scientifique :1.90e-3.0

Dans le programme l’affichage choisi est :



Si on ne veut pas utiliser l’écriture scientifique, il suffit de taper à la place :



**4. Utilisation du programme pour une autre réaction :**

On s’intéresse maintenant à la réaction entre 2 moles de dioxygène gazeux et 3 moles de dihydrogène gazeux qui forme de l’eau.

**1.** Ecrire l’équation modélisant la réaction. O2 + H2 -> 2 H2O

**2.** Quelles sont les parties du programme Python qu’il faudra modifier.

*Parties :*

- Quantités de matière initiales

- Nombres stoechiométriques

- Formules des réactifs et produits

***Ainsi que toutes les lignes où figure « Produit2 » (mettre un dièse devant la ligne correspondante)***

La partie « Recherche de Xmax » n’est pas à modifier

**3.**

**Voir le code corrigé : 1ere\_Etat\_final\_reaction\_Cor\_exo2.py**

**5.** Il suffit de modifier les quantités initiales dans la partie du programme : «**Quantités de matière initiales**»

# Programme Python permettant de déterminer l'état final d'un système chimique

# pour une réaction du type Réactif1 + Réactif2 -> Produit1 + Produit2

# avec coefficients stœchiométriques pouvant être différents.

# Code à destination des élèves qui peuvent modifier le code sous les parties roses

# Code volontairement peu optimisé (pas d'utilisation de listes) afin d'avoir

# une approche très simple pour les élèves non familiers avec le codage

# ##########################################################################

# O. CHAUMETTE - Lycée JP SARTRE - 69500 BRON - olivier.chaumette@ac-lyon.fr

# version 1.1. du code - février 2019

# ##########################

# Quantités de matière initiales

""" Entrer ci-dessous les quantités de matières initiales en moles"""

# n1\_ini n2\_ini pour les réactifs,n3\_ini et n4\_ini pour produits. Variables de type float (réelles)

# réactifs:

n1\_ini **=**2e-2

n2\_ini **=**4e-2

# produits

n3\_ini **=**0

n4\_ini **=**0

# ###########################

# Nombres stoechiométriques

""" Entrer ci-dessous les nombres stoechiometriques"""

# stoechio: variables de type "nombre entier" contenant les coef stoechiométriques

# réactifs:

stoechio1**=**2

stoechio2**=**3

# produits

stoechio3**=**2

stoechio4**=**3

# ##########################

# Formules des réactifs et produits

""" Entrer ci-dessous les formules des espèces chimiques"""

# Variables de type "texte" contenant les formules, non mises en formes, des espèces.

# réactifs:

r1**=**"Al"

r2**=**"Cu2+"

# produits

p1**=**"Al3+"

p2**=**"Cu"

# #########################

# initialisation des variables

r\_limit**=**"" # contiendra le nom du réactif limitant

xmax**=**0

# Quantité de matières finales:

n1\_fin **=**0

n2\_fin **=**0

n3\_fin **=** 0

n4\_fin **=** 0

# ###########################

# Affichage de l'état initial du système

# mise en forme de l'affichage, En écriture scientifique, à 2 CS

# "%.1e"% signifie écriture scientifique à 1 chiffre après la virgule (donc 2 C.S.)

**print(**"Etat initial du système:"**)**

**print(**"n("**+** r1 **+**") = "**+**"%.1e"**%**n1\_ini **+** " mol"**)**

**print(**"n("**+** r2 **+**") = "**+**"%.1e"**%**n2\_ini **+** " mol"**)**

**print(**"n("**+** p1 **+**") = "**+**"%.1e"**%**n3\_ini **+** " mol"**)**

**print(**"n("**+** p2 **+**") = "**+** "%.1e"**%**n4\_ini **+** " mol"**)**

**print(**"-------------------------------------"**)**

# ########################################

# recherche du réactif limitant et de xmax

""" Taper ici le code permettant de calculer xmax.

mettre dans la variable "r\_limit" le nom du réactif limitant """

**if** n1\_ini**/**stoechio1 **<** n2\_ini**/**stoechio2**:**

 # si celui du 1er réactif est plus petit, c'est lui le limitant et on calcule xmax

 xmax **=**n1\_ini**/**stoechio1

 # on mets dans react\_limit le nom du reactif limitant

 r\_limit**=** r1

**else:**

 xmax **=** n2\_ini**/**stoechio2

 r\_limit**=** r2

# ######################################

# Affichage du réactif limitant et de xmax

**print(**"Le réactif limitant est "**+**r\_limit**)**

**print** **(**"xmax="**+**"%.1e"**%**xmax**+**" mol"**)**

**print(**"----------------------------"**)**

# ###########################################

# Calcul des quantités de matière finales

""" Saisir ci-dessous les formules permettant le calcul des quantités

de matières finales en utilisant les bonnes variables """

# formules issues de la dernière ligne d'un tableau d'avancement

n1\_fin **=**n1\_ini **-**stoechio1**\***xmax

n2\_fin **=**n2\_ini **-**stoechio2**\***xmax

n3\_fin **=**n3\_ini **+**stoechio3**\***xmax

n4\_fin **=**n4\_ini **+**stoechio4**\***xmax

# ###########################

# Affichage de l'état final du système

**print(**"Etat final du système:"**)**

**print(**"n("**+** r1 **+**") = "**+**"%.1e"**%**n1\_fin **+** " mol"**)**

**print(**"n("**+** r2 **+**") = "**+**"%.1e"**%**n2\_fin**+** " mol"**)**

**print(**"n("**+** p1 **+**") = "**+**"%.1e"**%**n3\_fin **+** " mol"**)**

**print(**"n("**+** p2 **+**") = "**+**"%.1e"**%**n4\_fin **+** " mol"**)**

**print(**"-------------------------------------"**)**