

Fiche méthode

Référentiel, compétences

Capacités numérique : Déterminer la composition de l'état final d'un système chimique, siège d'une transformation chimique totale à l'aide d'un langage de programmation.

Commentaires de l'auteur

Les élèves doivent être déjà familiarisés avec l'environnement python de la calculatrice. De plus, ils doivent déjà connaître, en langage python :

- Les types simples, dont les booléens.
- Les structures de données complexes : listes et dictionnaires.
- Les fonctions.
- Les boucles bornées (parcours dans un dictionnaire) et non bornées (while).
- Les conditions.

Pour la partie chimie : on suppose que la définition de l'avancement chimique est connue.

Materiel

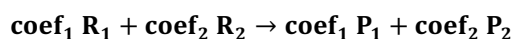
- Calculatrice TI-83 Premium CE Edition Python.
- Un ordinateur muni d'un IDE python quelconque pour saisir le programme (facultatif).

Enoncé

On réalisera un programme qui permettra de déterminer l'état final d'un système chimique.

Ce programme doit pouvoir s'adapter à toute transformation chimique dont on connaît l'équation, et les quantités de matière initiales (en mol) pour les réactifs et les produits (*les conditions initiales*).

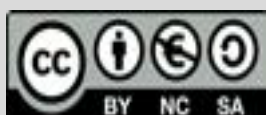
Pour renseigner l'équation et les conditions initiales on utilisera une structure de données de type dictionnaire. Soit une réaction chimique d'équation :



Si les réactifs R_i sont en quantité de matière initiale nr_i , et les produits P_i en quantité de matière np_i : Le format de données sera le suivant :

```
eq = {"reac":{"R1" : [coef1, np1],"R2" : [coef2,nr2]},  
      "prod":{"P1" : [coef1, np1],"P2" : [coef2,np2]}}
```

Le programme devra contenir les fonctions nécessaires au calcul de l'avancement maximum x , et modifier les valeurs nr_i et np_i du dictionnaire eq .



Thème: transformation de la matière :

Suivi et modélisation de l'évolution du système chimique

Proposition de résolution

Numériser l'équation et le système chimique

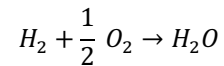
Dans l'éditeur python de la calculatrice : créer un nouveau script que l'on appellera : **AVAN**

On commence par initialiser le dictionnaire **eq** à partir de ses clés "**react**" et "**prod**". Dans le script **AVAN**, saisir :

```
eq = { "react": {}, "prod": {} }
```

Pour utiliser les fonctions du script, il faudra commencer par remplir les valeurs pour les clés "**react**" et "**prod**".

Prenons un exemple : Soit l'équation de la réaction de synthèse de l'eau :



Les coefficients associés à chacune des espèces chimiques sont :

- 1 pour H_2
- 0.5 pour O_2
- 1 pour H_2O .

Supposons les quantités de matière, dans l'état initial égales à

- 2 mol pour H_2
- 2 mol pour O_2
- 0 mol pour H_2O .

Pour représenter cette équation de réaction et le système chimique dans l'état initial, on devra alors écrire :

⇒ Pour les réactifs :

```
eq["react"] = { "H2": [1,2], "O2": [0.5,2] }
```

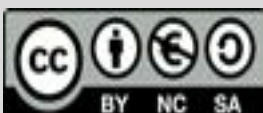
⇒ Pour le(s) produit(s) :

```
eq["prod"] = { "H2O": [1,0] }
```

La valeur associée à la clé "**react**" est-elle même un dictionnaire, où chacune des clés est la référence à l'une des espèces chimiques, parmi les réactifs. Ces clés sont uniques.

La valeur associée à chaque clé "**H2**" et "**O2**" est une liste comprenant 2 valeurs :

- La première valeur, au **rang 0** de la liste, contient la valeur numérique du **coefficient stœchiométrique** (ici, on mettra 1 pour "**H2**" et 0.5 pour "**O2**").
- La deuxième valeur, au **rang 1**, contient le **nombre de mole** de l'espèce dans l'état initial.



Thème: transformation de la matière :

Suivi et modélisation de l'évolution du système chimique

Pour le produit de réaction, à la clé "H2O", le coefficient stœchiométrique étant égal à 1, et sa quantité de matière à l'état initial sera nulle. On met alors [1,0] pour valeur.

Une autre possibilité serait de renseigner directement le contenu du dictionnaire **eq** avec l'instruction unique suivante :

```
eq = {"react":{"H2": [1,2], "O2": [0.5,2]}, "prod":{"H2O": [1,0]}}
```

Le tableau d'avancement à l'état initial sera le suivant :

equation	H ₂	+	0,5 O ₂	=	H ₂ O
avancement	react["H2"][1]		react["O2"][1]		prod["H2O"][1]
x = 0 mol	2		2		0

Fonction fin

La fonction **fin** détermine si le système est à l'état final (alors elle retourne **True**) ou s'il peut encore « avancer » (retourne **False**).

Cette fonction prend un paramètre, **react**. C'est une variable qui devra contenir la valeur associée à la clé "react" du dictionnaire **eq**.

Dans le script, on saisira :

```
def fin(react):  
    b=True  
    if react!={}:  
        b=False  
        for i in react.keys():  
            if react[i][1]<=0:  
                b = True  
    return b
```

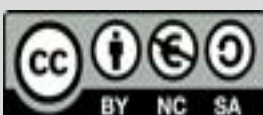
Le programme parcourt tous les éléments (toutes les clés) du dictionnaire **react**, quel que soit le nombre d'espèces chimiques, à condition qu'une espèce chimique au moins ait été renseignée.

A chaque itération (pour chacun des réactifs), la valeur du nombre de mol est comparée à 0. Si cette valeur est inférieure ou égale à zéro, la fonction renvoie True (pour marquer la fin de la réaction chimique).

Fonction avance

La fonction **avance** prend en paramètres :

- La variable **eq**, qui contient les paramètres de l'équation chimique et de l'état du système chimique.
- Une valeur **dx** qui représente le pas avec lequel on fera croître l'avancement **x** à chaque itération.



Thème: transformation de la matière :

Suivi et modélisation de l'évolution du système chimique

Cette fonction va modifier la valeur des quantités de matière pour chacune des espèces (valeur des listes "react" et "prod" au rang 1) et retourne l'avancement x lorsque le premier réactif arrive à zéro.

A la suite du programme, saisir la fonction suivante :

```
def avance(eq,dx):  
    x=0  
    reac = eq["react"]  
    prod = eq["prod"]  
    while fin(reac)==False: # il  
        reste des reactifs  
        x+=dx  
        for i in reac.keys():  
            reac[i][1]-=reac[i][0]*dx  
        for j in prod.keys():  
            prod[j][1]+=prod[j][0]*dx  
    return x
```

Tant qu'il reste des réactifs (c'est-à-dire `while fin(reac)==False`), alors :

- L'avancement est augmenté de dx avec : $x+=dx$
- Pour les réactifs, la quantité de matière est diminuée de $dx \times \text{coefficient_stœchiométrique}$:
- Pour chaque produit, la quantité de matière est augmentée de $dx \times \text{coefficient_stœchiométrique}$:

```
reac[i][1]-=reac[i][0]*dx
```

```
prod[j][1]+=prod[j][0]*dx
```

```
return x
```

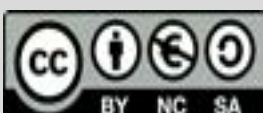
La fonction retourne finalement la valeur de l'avancement *maximum*.

Utiliser le programme

Exécuter le programme. Puis, dans le shell, écrire les instructions pour numériser le système chimique et son équation. Pour une écriture plus rapide, on écrit une clé numérique à la place des formules chimiques des espèces. L'exemple précédent devient alors :

⇒ Pour les réactifs :

```
PYTHON SHELL  
>>> # Shell Reinitialized  
>>> # L'exécution de PYTHON01  
>>> from PYTHON01 import *  
>>> eq["react"]={1:[1,2],2:[0.5,2]}  
}]  
Fns... | a A # | Outils | Éditer | Script
```



Thème: transformation de la matière :

Suivi et modélisation de l'évolution du système chimique

⇒ Puis pour le produit :

```
>>> eq["prod"]={1:[1,0]}
```

Appeler alors la fonction **avance** avec pour paramètres, la variable **eq**, et le pas égal à **0.1**

```
>>> avance(eq,0.1)
2.0
```

La fonction retourne alors la valeur de l'avancement maximum, pour le système chimique à l'état final : **2.0** mol.

On peut également consulter l'état final du système, en tapant **eq** :

```
>>> eq
{'react': {2: [0.5, 1.0], 1: [1,
-6.38378239159465e-16]}, 'prod'
{1: [1, 2.0]}}
```


Ce qui peut être représenté par le tableau d'avancement suivant :

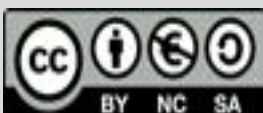
equation	H ₂	+	0,5 O ₂	=	H ₂ O
avancement	react["H2"][1]		react["O2"][1]		prod["H2O"][1]
x = 0 mol	2		2		0
x mol	2-x		2-0,5*x		x
x = 2.0 mol	0.0		1.0		2.0

Ecriture du programme dans un autre editeur

Le programme complet est relativement court, mais si l'on souhaite y ajouter des commentaires, il peut être plus pratique d'utiliser un IDE python pour le saisir, puis l'envoyer vers la calculatrice à l'aide du logiciel TI Connect.

Pour réaliser cette manipulation :

- Enregistrer le programme une fois celui-ci saisi dans l'ordinateur. Par exemple avec le nom *avancement.py*.
- Connecter la calculatrice à l'ordinateur. Lancer TI Connect.
- Appuyer sur le bouton  (Ajouter les données de l'ordinateur à la ou aux calculatrices connectées)
- Débrancher alors la calculatrice et lancer son application python. Chercher alors le programme qui vient d'être distribué à la calculatrice. Le nom a certainement été modifié lors du transfert ; il a peut-être été changé en PYTHON01



Ce document est mis à disposition sous licence Creative Commons <http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>

© Texas Instruments 2020 / Photocopie autorisée

[Date]

Thème: transformation de la matière :

Suivi et modélisation de l'évolution du système chimique

Script AVAN complet

```
ÉDITEUR : PYTHON01
LIGNE DU SCRIPT 0002
eq={"reac":{}, "prod":{}}

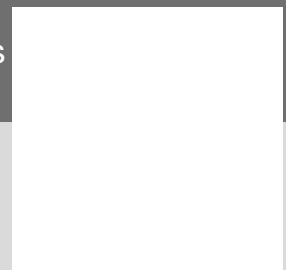
def fin(reac):
    b=True
    if reac!={}:
        b=False
        for i in reac.keys():
            if reac[i][1]<=0: b
                = True
    return b

def avance(eq,dx):
    x=0
    reac = eq["reac"]
    prod = eq["prod"]
    while fin(reac)==False: # il
        reste des reactifs
        x+=dx
        for i in reac.keys():
            reac[i][1]-=reac[i][
                0]*dx
        for j in prod.keys():
            prod[j][1]+=prod[j][
                0]*dx
    return x
```

Fns...	a A #	Outils	Exéc	Script
--------	-------	--------	------	--------

Documents et script à télécharger à l'adresse :
<https://education.ti.com/fr/physique-chimie>

Pour profiter de tutoriels vidéos, Flasher le QRCode ou cliquer dessus



Ce document est mis à disposition sous licence Creative Commons <http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>
© Texas Instruments 2020 / Photocopie autorisée