Capacité numérique

Tracer, à l'aide d'un langage de programmation, le taux d'avancement à l'équilibre en fonction de la température pour un système siège d'une transformation chimique modélisée par une seule réaction.

Etude du procédé Deacon

En présence d'un catalyseur à base de sulfate ou de chlorure de cuivre déposé sur de la pierre ponce, le dichlore peut être préparé vers 800 K par oxydation du chlorure d'hydrogène selon l'équilibre de Deacon :

$$4\ HCl_{(g)} + O_{2(g)} = 2\ Cl_{2(g)} + 2\ H_2O_{(g)}$$

Industriellement la synthèse est réalisée à température et pression constantes à partir d'un mélange des réactifs en proportions stœchiométriques.

Données :

- Constante des gaz parfaits : $R = 8,314\ J \cdot K^{-1} \cdot mol^{-1}$;
- Pression totale à laquelle la réaction est réalisée : P = 1 bar.
- Grandeurs thermodynamiques standard à 298 K :

Composé	$HCl_{(g)}$	$H_2O_{(g)}$	$Cl_{2(g)}$	$O_{2(g)}$
$\Delta_f H°$ (kJ.mol^{-1})	- 92,8	- 241,8		
S_m° (J.K^{-1}.mol^{-1})	186,9	188,7	223,1	205,1

I - Taux d'avancement et taux de conversion d'un réactif

1. Rappeler la relation de définition du taux d'avancement et du taux de conversion d'un réactif . Dans quel cas ces deux grandeurs ont-elles la même valeur ?

2. Exemple d'application : on suppose qu'au cours d'un essai, on introduit initialement dans le réacteur un mélange de 1 mole de HCl et de 1 mole de O_2. Dans l'état final , la quantité de matière de HCl est de 0,6 mol. Déterminer dans cet état final le taux d'avancement, ainsi que le taux de conversion de HCl.

3. Dans toute la suite, on se place dans les conditions industrielles. Établir le bilan de matière dans l'état final en fonction du taux d'avancement final τ.

II - Constante d'équilibre

4. Déterminer la valeur de la constante d'équilibre à $T_0 = 298$ K, soit $K°(T_0)$.

5. Trouver une expression littérale de la constante d'équilibre en fonction de la température $K°(T)$.

6. Établir l'expression littérale du quotient réactionnel en fonction de la pression totale P fixée et du taux d'avancement τ dans le cadre des conditions industrielles retenues.

7. À pression constante, écrire la condition d'équilibre chimique sous la forme $f(\tau) = 0$.

III - Optimisation du procédé Deacon

8. À pression constante, prévoir qualitativement l'influence d'une augmentation de température sur la valeur de τ.

9. À température constante, prévoir qualitativement l'influence d'une augmentation de pression sur la valeur de τ.

Le but des questions suivantes est de confirmer numériquement ces prédictions.

IV - S'organiser pour réussir

10. On suppose que l'état final est un état d'équilibre.
Poser par écrit la suite d'opérations à effectuer pour être capable de tracer le graphe illustrant l'évolution du taux d'avancement final en fonction de la température. Aucune ligne de code n'est demandée.
Le but de cette question est que vous explicitiez **votre propre démarche**. Après avoir fait cela, vous devriez plus facilement répondre à la question suivante qui, elle, nécessite d'écrire du code à l'intérieur d'un programme déjà organisé.

V - Programmation en Python

11. Compléter le script Python afin d'obtenir le graphe représentant $\tau_{(\acute{e}q)} = g(T)$.
Vous pouvez compléter le script par écrit, ou en modifiant directement le Google Colab proposé.

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import bisect

# Données
R=8.314                         # Constante des GP en uSI
DH0=                            # Noter la valeur de l'enthalpie standard de réaction à 298K en J/mol
DS0=                            # Entrer la valeur de l'entropie standard de réaction à 298K en J/K/mol
P =                             # Entrer la valeur de la pression totale en bar

# Valeur de la constante d'équilibre à 298K
K298=
print (K298)

# T : vecteur contenant 100 valeurs de température entre 298 et 1000 K
T=np.linspace( 298,1000,100)

# Valeurs de la constante d'équilibre en fonction de T
def K_zero(t):
    return

# Valeur du taux d'avancement  pour T = 298 K
def Tau_298(x):
    return
tau0=bisect(Tau_298,0.001,0.999)
print  ("la valeur du taux d'avancement à 298K est : ",tau0)

# Valeurs du taux d'avancement en fonction de la température et graphe tau (T)
Y0=np.zeros(len(T))
def Tau_T(x,t):
    return
for i in range (len(T)):
    Y0[i]=bisect(Tau_T,0.001,0.999,args=T[i])

plt.figure(0)
plt.plot(T,Y0,'+g')
plt.grid()
plt.title("tau en fonction de la température T")
plt.xlabel("T(K)")
plt.ylabel("tau")
plt.show ()
```

VI - Exploitation du graphe obtenu

12. Le graphe obtenu est-il en accord avec la prévision faite à la question **8**. ?

13. Industriellement, la température est choisie entre 700 et 900 K. Commentez ce choix.

VII - Influence de la pression

14. Compléter le script Python précédent pour tracer l'évolution du taux d'avancement en fonction de la température pour deux pressions différentes. Le faire pour P = 1 bar et P = 10 bars .

15. Le graphe obtenu est-il en accord avec la prévision faite à la question **9**. ?

16. Commenter le choix industriel de pression.

Utilisation de la méthode `bisect`

Pour une fonction **d'une variable** x, par exemple $f(x) = x^2 - 2$, la méthode `bisect` permet de trouver la racine sur un intervalle prédéterminé par la méthode de la dichotomie (qu'il est donc inutile de reprogrammer). Si on cherche la racine de $f(x)$ sur l'intervalle [1,2], alors le code approprié est :

```
def f(x):
    return x**2-2
sol=bisect(f,1.,2.)
print ( "la racine est : ",sol)
```

Il est retourné :

```
la racine est :  1.4142135623715149
```

Notons que, pour le langage Python, `x` est un argument pour la fonction `f`.

Pour une fonction **d'une variable** x, et **plusieurs paramètres** (ici a, b), par exemple $f\_bis(x, a, b) = \left(\frac{x}{a} + b\right)^2 - 2$, la méthode `bisect` doit inclure les valeurs des paramètres a et b. Si on cherche la racine de $f\_bis(x, a, b)$ sur l'intervalle [-1,1], avec les paramètres a = 0.7 et b = 2.3 alors le code approprié est :

```
def f_bis(x,a,b):
    return (x/a+b)**2-2
sol=bisect(f_bis,-1.,1.,args=(0.7,2.3))
print ( "la racine est : ",sol)
```

Il est retourné :

```
la racine est :  -0.6200505063370656
```

Notons que, pour le langage Python, `(x,a,b)` sont des arguments pour la fonction `f_bis`.

Lien et QR code vers la vidéo de la solution

https://www.youtube.com/watch?v=ekKN4C3S9-Y