

Capacité numérique : taux d'avancement à l'équilibre en fonction de la température

Objectif

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. Durée = 1h.

I. Etude préliminaire

On considère la synthèse du trioxyde de soufre selon l'équation bilan :



On part d'un mélange stoechiométrique contenant 1 mol de dioxygène. La pression P est égale à la pression standard $P^\circ = 1\text{bar}$.

On note ξ l'avancement réactionnel molaire de la réaction et ξ_{eq} l'avancement réactionnel molaire à l'équilibre.

On donne : $\ln K^\circ(T) = 198000/RT - 188/R$.

Q1. Déduire de la donnée sur $\ln K^\circ$, la valeur de l'enthalpie standard de réaction $\Delta_r H^\circ$ et la valeur de l'entropie standard de réaction $\Delta_r S^\circ$. On se placera dans l'approximation d'Ellingham que l'on rappellera.

Q2. Dresser le tableau d'avancement de la réaction et exprimer le taux de conversion à l'équilibre a en fonction de x_{eq} .

Q3. Exprimer le quotient réactionnel Q en fonction de x , P et P° .

Q4. Que vaut Q à l'équilibre chimique ? Donner l'expression de K° en fonction de ξ_{eq} , P et P° .

Q5. Rappeler le principe de la dichotomie. Quelle fonction faut-il choisir pour déterminer l'avancement réactionnel ξ_{eq} à l'équilibre ?

Données

$$R = 8,31 \text{ J}\cdot\text{K}^{-1}\cdot\text{mol}^{-1}$$

II- Programmation

Q6. Introduire dans le programme python la définition de la fonction notée constante(T), renvoyant la valeur de la constante d'équilibre K° à la température T.

Q7. Introduire dans le programme python la définition de la fonction notée quotient(ksi), renvoyant la valeur de la constante du quotient réactionnel en fonction de l'avancement réactionnel ksi.

Q8. Compléter le programme permettant de rechercher par dichotomie l'avancement réactionnel `avancement_eq(T)`.

Q9. Remplir la liste « ksi » donnant les valeurs de l'avancement réactionnel à l'équilibre `avancement_eq(T)` en fonction de T.

Q10. Compléter le programme python pour tracer `ksi(T)`.

Q11. Tracer le graphe de l'évolution de l'avancement à l'équilibre en fonction de la température. L'influence de la température sur la position de l'équilibre était-elle prévisible ? Justifier.

Annexe 1 : programme Python à compléter

```
import matplotlib.pyplot as plt # fonctions de trace
from math import exp # fonction exponentielle necessaire

#equation de la réaction : 2 SO2(g) + O2(g) = 2 SO3(g)
#quantités initiales : 2 mol de dioxyde de soufre et 1 mol de dioxygène
#P = P°

def constante(T):
    # renvoie la valeur de la constante thermodynamique d'equilibre à la
    temperature T
    A COMPLETER

def quotient(ksi):
    # renvoie la valeur du quotient reactionnel pour l'avancement ksi a
    la pression standard et pour des quantites de matiere initiales de 1
    mol de dioxygene et 2 mol de dioxyde de soufre (proportions
    stoechiometriques)
    A COMPLETER

def avancement_eq(T):
    #renvoie l'avancement de la reaction a l'equilibre en cherchant par
    dichotomie la solution de l'equation Q - K = 0

    # Initialisation des variables
    gauche = 0.000001 # xi minimal nul (mais le quotient reactionnel
    diverge en zero)
    droite = 0.999999 # xi maximal = 1 (idem)
    ksi = 0.5 # on commence par tester l'avancement ksi = 0.5 mol
    K = constante(T) # valeur de la constante thermodynamique d'equilibre
    a T

    # Premiere valeur du quotient reactionnel (pour un avancement de 0.5
    mol, au milieu de l'intervalle)
    Q = quotient(ksi)

    while abs(Q-K) > 0.0001: # condition d'arret
        if Q > K: # si le quotient reactionnel est superieur a K,
            l'avancement a l'equilibre est a chercher a gauche
            droite = ksi
        else:
            gauche = ksi # sinon on cherche a droite
            ksi = A COMPLETER # nouveau milieu de l'intervalle
            Q = quotient(ksi) # nouvelle valeur de Q pour le milieu
            l'intervalle
    return ksi
```

```
## Corps principal
# Trace en fonction de la temperature entre 600 et 1500 K (un point tous
les 10 K)

T = [temp for temp in range(600,1500,10)] # liste des abscisses
ksi = [] # liste vide des ordonnees
for temp in T:
    A COMPLETER # ajout successif des valeurs de l'avancement a
    l'equilibre dans la liste des ordonnees

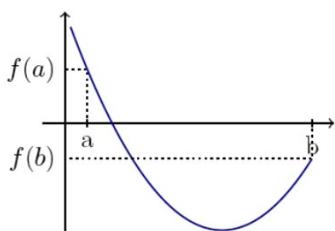
A COMPLETER # trace du nuage de points

# titres des axes et du graphique
plt.ylabel("avancement a l'equilibre (mol)")
plt.xlabel("temperature (K)")
plt.title("Influence de la temperature sur l'oxydation de SO2")
plt.show()

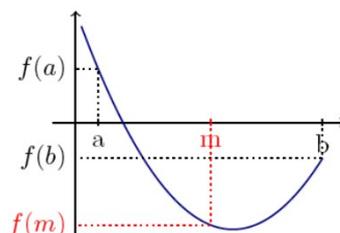
# enregistrement du graphique dans un fichier image
plt.savefig('oxydation.png')
```

Annexe 2 : rappel sur la recherche de zéro par la méthode de la dichotomie

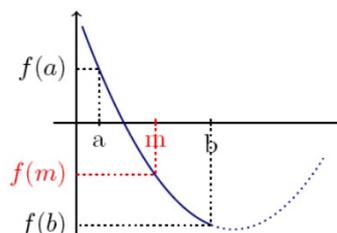
C'est une méthode relativement simple et efficace pour trouver l'unique zéro d'une fonction $f(x)$ sur un intervalle $[a, b]$ donné avec une précision eps souhaitée.



La fonction ne s'annulant qu'une seule fois sur l'intervalle d'étude, on doit avoir $f(a) \times f(b) < 0$. Considérons le cas de figure ci-contre.



On prend m , milieu de $[a, b]$ et on calcule $f(m)$



Ici la fonction s'annule entre a et m .

Il suffit donc de réitérer le processus en prenant comme nouvelle valeur de b la valeur de m .

On continue ainsi de suite tant que $b - a > eps$.

Un programme implémentant la méthode de dichotomie a, typiquement, la structure suivante :

```
1 def fTest(x):
2     return x*x - 2
3
4 def dichotomie(f, a, b, eps) :
5     while b - a > eps :
6         m = (a+b)/2 # milieu de [a,b]
7         if f(a)*f(m) <= 0 :
8             b = m
9         else:
10            a = m
11     return m
12
13 solution = dichotomie(fTest, 0, 4, 0.001)
14 print(solution)
```