Capacité numérique : détermination de grandeurs standard d'activation

Objectif

Déterminer, à l'aide d'un langage de programmation, les grandeurs standard d'activation d'une réaction en réalisant une régression linéaire. **Durée = 1h**.

I. Etude préliminaire

La réaction d'hydrolyse du chlorure de tertiobutyle dans un mélange eau-éthanol est étudié à plusieurs températures :

$$(CH_3)_3Cl + H_2O = (CH_3)_3OH + Cl^- + H^+.$$

Les mesures de la constante de réaction en fonction de la température sont reportées dans le tableau suivant :

T (K)	292	298	303	308	313
k (10^{-4}s^{-1})	1,48	3,03	4,50	8,17	12,8

On cherche à déterminer les grandeurs standard d'activation en utilisant la relation d'Eyring, qui s'écrit pour une réaction d'ordre global égal à 1 :

$$k = \frac{k_B T}{h}\exp\left(-\frac{\Delta_r G^{o\neq}}{RT}\right)$$

soit :

$$k = \frac{k_B T}{h}\exp\left(-\frac{\Delta_r H^{o\neq} - T\Delta_r S^{o\neq}}{RT}\right) = \frac{k_B T}{h}\exp\left(\frac{\Delta_r S^{o\neq}}{R}\right)\exp\left(-\frac{\Delta_r H^{o\neq}}{RT}\right)$$

Q1. Quel est le mécanisme mis en jeu pour la réaction ? Justifier et écrire le mécanisme.

Q2. Quelle est l'étape cinétiquement déterminante ? En déduire la loi cinétique de la réaction et son ordre global. Confirmer le résultat en observant l'unité de k dans le tableau.

Q3. Rappeler la signification des différentes grandeurs dans la relation d'Eyring et vérifier son homogénéité.

Q4. Linéariser la relation d'Eyring et expliquer quelle fonction f étudier pour déterminer les grandeurs standard d'activation par régression linéaire.

Données

R = 8,31 J.K^{-1}.mol^{-1}.

II- Programmation

Q5. Introduire dans le programme python la liste des valeurs de température T et la liste des valeurs de constante cinétique k.

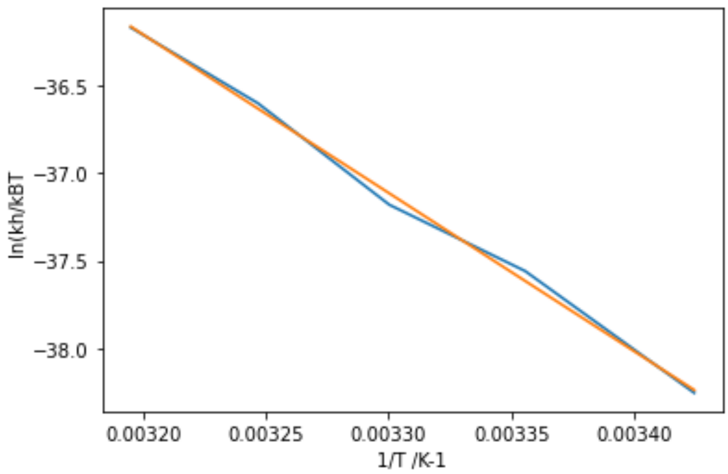
Q6. Coder en Python la fonction f à étudier.

Q7. Utiliser le module polyfit de numpy pour calculer la pente et l'ordonnée à l'origine de la droite de régression. Tracer la droite de régression.

Q8. En déduire l'enthalpie standard d'activation et l'entropie standard d'activation.

Q9. En quoi le signe de l'entropie standard d'activation est-il en accord avec un mécanisme de type S_N1 ?

On obtient la représentation graphique suivante :



Annexe 1 : programme Python à compléter

```
import numpy as np
import matplotlib.pyplot as plt


# Données relatives au problème
kB = 1.380649e-23 #constante de Boltzmann en J/K
h = 6.62607015e-34 #constante de Planck en J.s

T = [] #liste des valeurs de température en K A COMPLETER

k = [] #Liste des valeurs de k en s-1 A COMPLETER


# Définition de la fonction log A COMPLETER
def f(k,T):
    return

# Liste par compréhension des inverses des températures
InverseT=[1/x for x in T]

# Liste par compréhension des ordonnées
Y=[f(k[i],T[i]) for i in range (5)]

# utilisation de polyfit d'ordre 1 pour faire la régression linéaire sur Y en
fonction de InverseT A COMPLETER

a,b =

print(a,b) # afficher les valeurs du coefficient directeur et de l'ordonnée à
l'origine

# Définition de la droite de régression A COMPLETER

Reglin=[]


# Tracé de Y = f(InverseT)
plt.plot(InverseT,Y)
plt.plot(InverseT,Reglin)
plt.xlabel('1/T /K-1')
plt.ylabel('ln(kh/kBT')
plt.show()
```

Corrigé : programme Python complété

```
import numpy as np
import matplotlib.pyplot as plt


# Données relatives au problème
kB = 1.380649e-23 #constante de Boltzmann en J/K
h = 6.62607015e-34 #constante de Planck en J.s

T = [292,298,303,308,313] #liste des valeurs de température en K

k = [1.48e-4, 3.03e-4, 4.5e-4, 8.17e-4, 12.8e-4] #Liste des valeurs de k en
s-1


# Définition de la fonction log
def f(k,T):
    return np.log(k*h/T/kB)

# Liste par compréhension des inverses des températures
InverseT=[1/x for x in T]

# Liste par compréhension des ordonnées
Y=[f(k[i],T[i]) for i in range (5)]

# utilisation de polyfit d'ordre 1 pour faire la régression linéaire sur Y en
fonction de InverseT

a,b = np.polyfit(InverseT, Y, 1)

print(a,b) # afficher les valeurs du coefficient directeur et de l'ordonnée à
l'origine

# Définition de la droite de régression

Reglin=[a/x+b for x in T]


# Tracé de Y = f(InverseT)
plt.plot(InverseT,Y)
plt.plot(InverseT,Reglin)
plt.xlabel('1/T /K-1')
plt.ylabel('ln(kh/kBT')
plt.show()
```