CN 2 : Influence de la température sur le taux d'avancement à l'équilibre : exemple de la synthèse de l'ammoniac

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.

L'exemple étudié est celui de la synthèse de l'ammoniac :

$$N_2(g) + 3\ H_2(g) = 2\ NH_3(g)$$

Conditions de la transformation étudiée et hypothèses :
- composition initiale : 1 mol de diazote et 3 mol de dihydrogène (proportions stoechiométriques) ;
- pression *P* fixée P= 5 bar;
- température *T* : on étudie la plage de température [300 K, 1200 K] ;
- l'ensemble du mélange est gazeux et se comporte comme un gaz parfait ;
- on se place dans l'approximation d'Ellingham.

Données thermodynamiques à 298 K :

	N_2 (g)	H_2 (g)	NH_3 (g)
$\Delta_f H°$ (kJ.mol^{-1})	0	0	- 45.9
S° (J.K^{-1}.mol^{-1})	192	131	193

1) Préparation théorique :

a) Déterminer l'évolution de ΔrG° en fonction de T
b) Rappeler la définition de K en fonction de ΔrG° et T
c) Exprimer le quotient de réaction en fonction du taux d'avancement τ.
☞ Faire un tableau d'avancement est toujours une très bonne idée !
d) Donner l'équation qui permet de déterminer τ_{eq} à l'aide de K.

2) Programme python

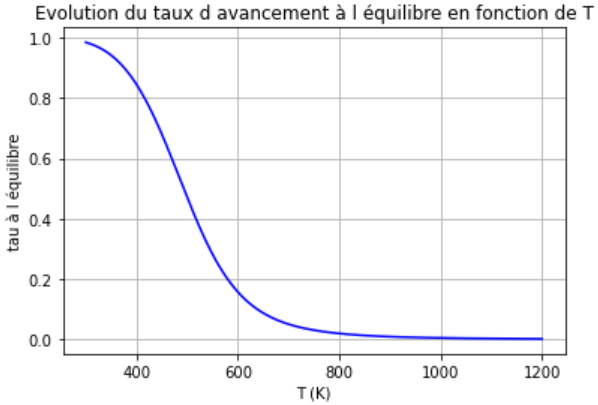
Le fichier ***CN2-AREMPLIR.py*** est disponible sur cdp onglet capacités numériques. Ce fichier contient l'appel des bibliothèques, les données numériques et la structure du programme. Il faut compléter les trous !!!

L'idée est de trouver les zéros d'une fonction issue de l'équation obtenue question d). Pour ce faire 3 possibilités au programme :

- la dichotomie (il faut au préalable déterminer, en s'appuyant sur une représentation graphique, un intervalle adapté à la recherche numérique d'une racine par la méthode dichotomique.)
- Utiliser une commande toute prête de python grâce au sous-module scipy.optimize (alias op) de la bibliothèque NumPy. On a 3 possibilités :

Fonctions	Actions réalisées
op.fsolve(f,xo)	Résoudre l'équation f(x) = 0, xo est la valeur initiale de l'algorithme
op.bisect(f,a,b)	Résoudre l'équation f(x) = 0 sur le segment [a ; b] par la méthode de dichotomie. Mettre la fonction f(x) sous forme polynomiale pour éviter les divisions par zéro
op.newton(f,xo)	Résoudre l'équation f(x) = 0 par la méthode de Newton en précisant la valeur initiale de l'algorithme

La structure du programme sera habituelle :
1- importation des bibliothèques
☞ ne pas oublier scipy.optimize
2- données de l'énoncé
3- fonction DrG0(T)
3- fonction K(T)
3- définition du polynôme à annuler f(x,T,P)
3- liste des abscisses
☞ commande np.linspace(a,b,N)
3- recherche des tau_eq pour chaque T, en utilisant op.newton
☞ faire une boucle
4-tracé de la courbe tau=f(T)



Si vous êtes arrivés à ce résultat :

☺☺☺☺ Félicitations, vous avez réussi votre seconde compétence numérique de 2nde année ☺☺☺☺

Sinon trouver vos erreurs ! votre programme DOIT compiler !!!