

CN3 – Détermination de point(s) de fonctionnement d'un RPA

Capacité numérique : à l'aide d'un langage de programmation, déterminer le(s) point(s) de fonctionnement (température et taux de conversion) d'un réacteur parfaitement agité continu siège d'une transformation modélisée par une réaction unique et en discuter la stabilité.

On considère la décomposition thermique du peroxyde de ditertiobutyle $t\text{Bu-O-O-}t\text{Bu}$ dans un réacteur parfaitement agité en régime stationnaire de volume $V = 500 \text{ mL}$ et de surface $S = 300 \text{ cm}^2$ alimenté en liquide pur avec un débit constant $D_v = 3 \text{ L.h}^{-1}$ à $T_e = 200 \text{ }^\circ\text{C}$.

La décomposition thermique du réactif peut être modélisée par une réaction d'ordre 1 et dont les paramètres d'Arrhenius sont : $A = 10^{15} \text{ s}^{-1}$ et $E_a = 157 \text{ kJ.mol}^{-1}$.

Le réacteur est refroidi par une double enveloppe maintenue à $T_0 = 20 \text{ }^\circ\text{C}$. Le coefficient de transfert conducto-convectif vaut $h = 80 \text{ W.m}^{-2}.\text{K}^{-1}$. On donne la loi de Newton : $\mathcal{P}_{th} = h(T_0 - T_s)S$.

Données

Masse molaire du réactif liquide : $M = 146 \text{ g.mol}^{-1}$

Masse volumique du réactif liquide : $\rho = 900 \text{ g.L}^{-1}$

Capacité thermique massique du réactif liquide : $c_{p,m}^o = 2,1 \text{ J.g}^{-1}.\text{K}^{-1}$

Enthalpie standard de réaction : $\Delta_r H^\circ = -150 \text{ kJ.mol}^{-1}$.

📄 Télécharger le script CN3-Indu-AREMPLIR.py (onglet chimie puis Capacités Numériques)

1. À partir d'un bilan de matière (BM), établir une équation reliant le taux de conversion en sortie X_a et la température T dans le réacteur. Coder sous Python une fonction $k(T)$ et une fonction $X_{a_BM}(T)$.
2. À partir d'un bilan thermique (BT), établir une autre expression reliant le taux de conversion en sortie X_a et la température dans le réacteur. Coder sous Python une fonction $X_{a_BT}(T)$.
3. Tracer les deux fonctions de sorte à visualiser graphiquement le(s) point(s) de fonctionnement. T varie de 200 à 700 K.
4. En utilisant la méthode de la dichotomie ou la fonction *bisect* de la bibliothèque *scipy.optimize*, déterminer les coordonnées du(des) point(s) de fonctionnement.
5. Étudier la stabilité du(des) point(s) de fonctionnement. Pour cela, coder sous Python :
 - une fonction $\text{deriv}(f,x,\text{epsy})$ qui renvoie la valeur de la dérivée d'une fonction f au point x , epsy étant le pas autour de x
 - une fonction $\text{stabil}(T_s)$ qui envoie « point stable » si un point de fonctionnement est stable et « point instable » s'il est instable.