

CAPACITÉ NUMÉRIQUE**CORRIGÉ**

Capacité numérique : Tracer, à l'aide d'un langage de programmation, l'évolution temporelle de la température pour un système siège d'une transformation adiabatique modélisée par une seule réaction chimique.

1. Réaction d'ordre 1 donc on a : $-\frac{d[A]}{dt} = k[A]$ Or ici la valeur de k dépend de T . On a donc : $\frac{d[A]}{dt} = -k(T) \cdot [A]$

2. À température constante, $k(T)$ est constante ; on a donc après séparation des variables et intégration entre $t = 0$ et t :

$$\ln [A]_t = \ln [A]_0 - k(T) \cdot t \quad \text{ou encore} \quad [A]_t = [A]_0 \cdot e^{-k(T) \cdot t}$$

On ne peut pas résoudre l'équation différentielle dans le cas présent car la température variant au cours du temps, $k(T)$ varie aussi.

3. Le volume étant constant, on a donc $\frac{d[A]}{dt} = \frac{1}{V} \frac{dn_A}{dt} = -k(T) \cdot \frac{n_A}{V}$ soit encore $\frac{dn_A}{dt} = -k(T) \cdot n_A$

Or on a le tableau d'avancement suivant :

	A	=	B
t = 0	n		0
t	n - ξ		ξ

On en déduit donc : $\frac{d(n-\xi)}{dt} = -k(T) \cdot (n-\xi) \Leftrightarrow -\frac{d\xi}{dt} = -k(T) \cdot (n-\xi)$ soit encore $d\xi = k(T) \cdot (n-\xi) \cdot dt$

4. Loi d'Arrhénius : $\frac{d(\ln k)}{dT} = \frac{E_a}{RT^2}$; après séparation des variables et intégration entre T_0 et T , on obtient :

$$\ln k(T) = \ln k(T)_0 - \frac{E_a}{R} \left(\frac{1}{T} - \frac{1}{T_0} \right) \quad \text{soit encore} \quad k(T) = k_0 \cdot e^{-E_a/R(1/T-1/T_0)}$$

5. On raisonne sur un intervalle de temps infinitésimal dt ; pendant cet intervalle de temps :

- l'avancement de la réaction évolue d'une quantité infinitésimale $d\xi$
- la température évolue d'une quantité infinitésimale dT
- la transformation étant adiabatique et à P constante, la variation infinitésimale d'enthalpie est donc : $dH = \delta Q_p = 0$

Tableau d'avancement lors de cette transformation infinitésimale :

	A	=	B
t	n - ξ		ξ
t + dt	n - ξ - dξ		ξ + dξ

On fait le même raisonnement que pour le calcul d'une température de flamme : on décompose en deux étapes fictives :

- Étape 1 : évolution de l'avancement à température constante T : alors $dH_1 = \Delta_r H^\circ \cdot d\xi$
 - Étape 2 : évolution de température à composition constante, caractérisée par un avancement $\xi + d\xi$; alors $dH_2 = C_p^\circ \cdot dT$
- Attention : pour le calcul de C_p° , il ne faut pas oublier la capacité thermique du réacteur. On a donc :

$$C_p^\circ = C + n_A \cdot C_{p,A}^\circ + n_B \cdot C_{p,B}^\circ = C + (n - \xi - d\xi) \cdot C_{p,A}^\circ + (\xi + d\xi) \cdot C_{p,B}^\circ$$

Or $dH = dH_1 + dH_2 = 0$; on en déduit donc : $\Delta_r H^\circ \cdot d\xi + (C + n_A \cdot C_{p,A}^\circ + n_B \cdot C_{p,B}^\circ - C + (n - \xi - d\xi) \cdot C_{p,A}^\circ + (\xi + d\xi) \cdot C_{p,B}^\circ) \cdot dT = 0$

soit encore
$$dT = \frac{-\Delta_r H^\circ d\xi}{C + C_{p,A}^\circ (n - \xi - d\xi) + C_{p,B}^\circ (\xi + d\xi)}$$

6. Il s'agit ici de faire le calcul classique de la température de flamme, cette fois en considérant la transformation macroscopique entre $t = 0$ et $t = +\infty$. Le tableau d'avancement est donc :

	A	=	B
$t = 0$	n		0
$t = +\infty$	0		n

En décomposant encore en deux étapes fictives, on a donc :

- Étape 1 : évolution de l'avancement à température constante T_0 : alors $\Delta H_1 = \Delta_r H^\circ \cdot (\xi_f - \xi_0) = \Delta_r H^\circ \cdot n$
- Étape 2 : évolution de température à composition constante (composition à $t = +\infty$) ; alors $\Delta H_2 = C_p \cdot (T_f - T_0)$ avec $C_p = C + n \cdot C_{p,B}^\circ$; alors $\Delta H_2 = (C + n \cdot C_{p,B}^\circ) \cdot (T_f - T_0)$

Avec $\Delta H = 0 = \Delta H_1 + \Delta H_2$ on en déduit :
$$T_f = T_0 - \frac{\Delta_r H^\circ}{C + n \cdot C_{p,B}^\circ}$$

7. Programmation Python : Script complété :

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3 from math import *
4
5 ## grandeurs liées au problème étudié
6 n = 1 # nombre initial de moles de A (mol)
7 T0 = 298 # T0 (K)
8 t0 = 0 # temps initial (s)
9 tF = 5000 # temps final (s)
10
11 ## grandeurs liées à la thermodynamique
12 DrH0 = -20000 # DrH° (J/mol)
13 cpA = 30 # Cp,A (J/K/mol)
14 cpB = 20 # Cp,B (J/K/mol)
15 C = 250 # capacité calorifique du réacteur (J/K)
16
17 ## grandeurs liées à la cinétique
18 R = 8.314 # constante des gaz parfaits (J/K/mol)
19 Ea = 20000 # énergie d'activation (J/mol)
20 k0 = 6e-4 # constante de vitesse à la température T0 (s-1)
21
22 ## on définit les listes qui vont recueillir les résultats désirés
23 Temps=[t0]
24 Temperature=[T0]
25 Ksi=[0]
26
27 ## on définit les paramètres utiles pour la résolution numérique
28 Npoints = 10000 # nombre d'itérations pour couvrir l'intervalle de temps [t0,tF]
29
30 # valeur de la constante de vitesse en fonction de la température
31 def k(T):
32     return k0*np.exp(-Ea/R*(1/T-1/T0))
33
34 # variation de ksi pendant l'intervalle de temps dt
35 def d_ksi(T,ksi,dt):
36     return k(T)*(n-ksi)*dt
37
38 # variation de la température T pendant l'intervalle de temps dt
39 def d_T(dksi,ksi):
40     return -dksi*DrH0/((n-ksi-dksi)*cpA + (ksi+dksi)*cpB + C)
41
42 # intervalle de temps entre deux estimations de la température
43 Delta_t = (tF-t0)/Npoints
44

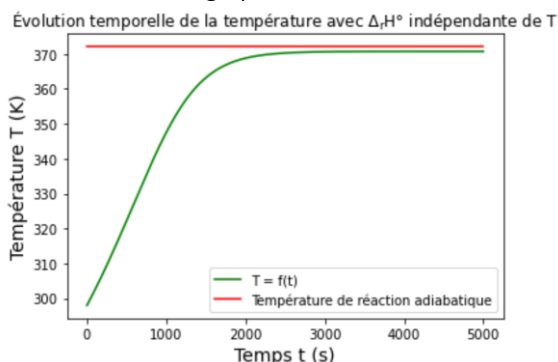
```

```

45 # Génération des listes t(temps), ksi et T(température)
46 for i in range(Npoints):
47     Temps.append(Temps[-1]+Delta_t)
48     dksi=d_ksi(Temperature[-1],Ksi[-1],Delta_t)
49     dT=d_T(dksi,Ksi[-1])
50     Ksi.append(Ksi[-1]+dksi)
51     Temperature.append(Temperature[-1]+dT)
52
53 # Calcul de la température adiabatique
54 T_adia=T0-n*DrH0/(n*cpB+C) ; print(T_adia)
55
56 # Tracé du graphe
57 plt.figure(1)
58 plt.plot(Temps,Temperature,'g',label="T = f(t)")
59 plt.plot([t0,tF],[T_adia,T_adia], 'r-',label="Température de réaction adiabatique")
60 plt.xlabel(r'Temps t (s)',fontsize=14)
61 plt.ylabel(r'Température T (K)',fontsize=14)
62 plt.title(r'Évolution temporelle de la température avec $\Delta_r H^\circ$ indépendante de T')
63 plt.legend()
64 plt.show()

```

8. On obtient donc le graphe suivant :



La courbe montrant l'évolution temporelle de la température est croissante, ce qui est attendu car la réaction est adiabatique. On observe cependant une légère différence entre la limite asymptotique de cette courbe et la température adiabatique. Remarque : ceci est dû à l'hypothèse d'Ellingham : on néglige la variation de $\Delta_r H^\circ$ avec la température.

9. En intégrant entre T_0 et T , on a donc : $\Delta_r H^\circ(T) = \Delta_r H^\circ(T_0) + (C_{p,B}^\circ - C_{p,A}^\circ) \cdot (T - T_0)$

10. On a alors :

```

35 def DrH0_T(T):
36     return DrH0 + (cpB-cpA)*(T-T0)

```

11. Nouveau code Python modifié pour tenir compte de la variation de $\Delta_r H^\circ$ avec T :

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3 from math import *
4
5 ## grandeurs liées au problème étudié
6 n = 1 # nombre initial de moles de A
7 T0 = 298 # T0 (K)
8 t0 = 0 # temps initial (s)
9 tF = 5000 # temps final (s)
10
11 ## grandeurs liées à la thermodynamique
12 DrH0 = -20000 # DrH° (J/mol)
13 cpA = 30 # Cp°A (J/K/mol)
14 cpB = 20 # Cp°B (J/K/mol)
15 C = 250 # capacité calorifique de l'enceinte (J/K)
16
17 ## grandeurs liées à la cinétique
18 R = 8.314 # constante des gaz parfaits en uSI
19 Ea = 20000 # énergie d'activation en J/mol
20 k0 = 6e-4 # constante cinétique à la température T0 (en s-1)
21

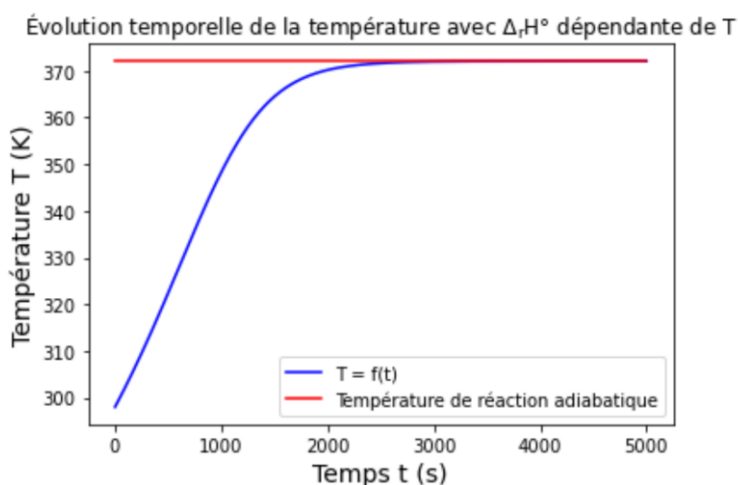
```

```

22  ## on définit Les listes qui vont recueillir Les résultats désirés
23  Temps=[t0]
24  Temperature=[T0]
25  Ksi=[0]
26
27  ## on définit Les paramètres utiles pour La résolution numérique
28  Npoints = 10000          # nombre d'itérations pour couvrir L'intervalle de temps [t0,tF]
29
30  # valeur de La constante de vitesse en fonction de La température
31  def k(T):
32      return k0*np.exp(-Ea/R*(1/T-1/T0))
33
34  # valeur de L'enthalpie standard de réaction en fonction de La température
35  def DrH0_T(T):
36      return DrH0 + (cpB-cpA)*(T-T0)
37
38  # variation de ksi pendant L'intervalle de temps dt
39  def d_ksi(T,ksi,dt):
40      return k(T)*(n-ksi)*dt
41
42  # variation de La température T pendant L'intervalle de temps dt
43  def d_T(dksi,ksi,T):
44      return -dksi*DrH0_T(T)/((n-ksi-dksi)*cpA + (ksi+dksi)*cpB + C)
45
46  # intervalle de temps entre deux estimations de La température
47  Delta_t = (tF-t0)/Npoints
48
49  # Génération des listes t(temps), ksi et T(température)
50  for i in range(Npoints):
51      Temps.append(Temps[-1]+Delta_t)
52      dksi=d_ksi(Temperature[-1],Ksi[-1],Delta_t)
53      dT=d_T(dksi,Ksi[-1],Temperature[-1])
54      Ksi.append(Ksi[-1]+dksi)
55      Temperature.append(Temperature[-1]+dT)
56
57  # Calcul de La température adiabatique
58  T_adia=T0-n*DrH0/(n*cpB+C) ; print(T_adia)
59
60  # Tracé du graphe
61  plt.figure(1)
62  plt.plot(Temps,Temperature,'b',label="T = f(t)")
63  plt.plot([t0,tF],[T_adia,T_adia],'r-',label="Température de réaction adiabatique")
64  plt.xlabel(r'Temps t (s)',fontsize=14)
65  plt.ylabel(r'Température T (K)',fontsize=14)
66  plt.title(r'Évolution temporelle de la température avec  $\Delta_r H^\circ$  dépendante de T')
67  plt.legend()
68  plt.show()

```

12. On obtient cette fois le graphe suivant :



On constate que cette fois, la limite asymptotique de la courbe $T = f(t)$ correspond bien à la température adiabatique.

FIN DU CORRIGÉ