CCP PC 18 Modélisation

1) $\rho V c_p \frac{dT}{dt} = \frac{dH}{dt}$ variation de l'enthalpie du système par unité de temps

P_{th} : puissance reçue par le système

$UA(T-T_p)$: puissance cédée par conduction-convection

Dimension : $[\rho V c_p] \sim [m][c_p]$

$[\rho V c_p\, dT] \sim [m][c_p][T] \sim$ [Energie]

$\left[\frac{\rho V c_p\, dT}{dt}\right] \sim \frac{[\text{energie}]}{[\text{temps}]} \sim [\text{Puissance}]$ ok

$[U] \sim \frac{[\text{Puissance}]}{[L]^2[T]}$ d'après son unité $W m^{-2} K^{-1}$

$[A] \sim [\text{surface}] \sim [L]^2$

$\rightarrow [U][A][T-T_p] \sim \frac{[\text{Puissance}][T][L]^2}{[L]^2[T]} \sim [\text{Puissance}]$ ok.

L'équation est un bilan thermique en puissance

2) En régime permanent rien ne dépend du temps ; d'où $P_{th} - UA(T-T_p) = 0$

$$\boxed{T-T_p = \frac{P_{th}}{UA}}$$

3) La figure 2a montre le régime permanent quand $T-T_p$ n'évolue plus avec le temps. On lit alors $\boxed{T-T_p = 40\,K}$

$\Rightarrow U = \frac{P_{th}}{A(T-T_p)} = \frac{96}{8 \cdot 10^{-3} \cdot 40} = 300\, W m^{-2} K^{-1}$

4) On repart de $\rho V c_p \frac{dT}{dt} = P_{th} - UA(T-T_p)$

qui s'écrit $\frac{dT}{dt} = \frac{P_{th}}{\rho V c_p} - \frac{UA}{\rho V c_p}(T-T_p)$

d'où $\boxed{s = \frac{P_{th}}{\rho V c_p}}$ et $\boxed{\tau_c = \frac{\rho V c_p}{UA}}$

5) Lors de la phase de refroidissement il n'y a plus P_{th} et l'équation devient : $\boxed{\frac{dT}{dt} = -\frac{T-T_p}{\tau_c}}$

6) Solution $\boxed{T-T_p = (T_{max}-T_p)\, e^{-\frac{t}{\tau_c}}}$

7) Enoncé : $\ln(\underset{\text{en Kelvin}}{T-T_p}) = -1{,}33 \cdot 10^{-2}\, \underset{\text{(secondes)}}{t} + 3{,}68$

Or la solution donne $\ln(T-T_p) = \ln(T_{max}-T_p) - \frac{t}{\tau_c}$

On en déduit $\tau_c = \frac{1}{1{,}33 \cdot 10^{-2}}$ secondes

soit $\tau_c = 75\, s$

D'après 4) on déduit $U = \frac{\rho V c_p}{A \tau_c} = \frac{10^3 \times 0{,}1 \times 10^{-3} \times 1800}{8 \cdot 10^{-3} \times 75}$

on retrouve $U = 300\, W m^{-2} K^{-1}$

8) $\boxed{r = -\frac{dC_R}{dt} = k C_R}$

r en $mol\, m^{-3} s^{-1}$ et k en s^{-1}

9) Loi d'Arrhénius : $\frac{d \ln k}{dT} = \frac{E_a}{RT^2}$

qui s'intègre en $\boxed{k = k_0 e^{-\frac{E_a}{RT}}}$

k et k_0 en s^{-1} ; E_a en $J \cdot mol^{-1}$

10) Ainsi $\boxed{\frac{dC_R}{dt} = \underbrace{-k_0 e^{-\frac{E_a}{RT}} C_R}_{f(C_R,T)}}$

11) A T = Cste : $\ln \frac{C_R}{C_{R_0}} = -k_0 e^{-\frac{E_a}{RT}} t$

$\rightarrow \boxed{C_R = C_{R_0} e^{-k_0 e^{-\frac{E_a}{RT}} t}}$

12) $\boxed{X_R = \frac{C_{R_0} - C_R}{C_{R_0}} = 1 - e^{-k_0 e^{-\frac{E_a}{RT}} t}}$

13) Si T_{max} constante alors $\frac{dC_R}{dt} = -k_{(T)} C_R$

Comme $X_R = \frac{C_R^\circ - C_R}{C_R^\circ}$ on obtient $C_R = C_R^\circ (1 - X_R)$

d'où $\boxed{+\frac{dX_R}{dt} = +k_0 e^{-\frac{E_a}{RT}} (1 - X_R)}$

14) $[\Delta_r H] \sim [\text{Energie}][\text{mol}]^{-1}$

$[r] \sim \dfrac{[\text{mol}]}{[L]^3[t]}$ et $V \sim [L]^3$

donc $[S] \sim [\Delta_r H][r][V] \sim \dfrac{[\text{énergie}]}{[t]} \sim [\text{Puissance}]$

$\boxed{\text{S est une puissance}}$

15) Système : le volume V

On aura le même bilan thermique équation (1) de l'énoncé, où il s'agit de déterminer P_{th} avec $\Delta_r H°$ de la réaction exothermique. ($\Delta_r H° < 0$)

$$P_{th} = S = -\Delta_r H° \, r \, V$$

avec $r = -\dfrac{dC_R}{dt}$ et $C_R = C_R°(1 - X_R)$

on a $r = + C_R° \dfrac{dX_R}{dt}$

donc $P_{th} = -\Delta_r H° C_R° \dfrac{dX_R}{dt} V$

donc en reprenant (1) :

$$\rho V c_p \frac{dT}{dt} = -\Delta_r H° C_R° \frac{dX_R}{dt} V - UA(T - T_p)$$

puis en divisant par $\rho V c_p$:

$$\frac{dT}{dt} = -\frac{\Delta_r H° C_R°}{\rho c_p} \frac{dX_R}{dt} - \frac{T - T_p}{\tau_c}$$

Ainsi $\boxed{J = -\dfrac{\Delta_r H° C_R°}{\rho c_p}}$ > 0 car $\Delta_r H° < 0$

16) Comme $\left[J \dfrac{dX_R}{dt}\right] \sim \dfrac{[T]}{[t]}$

on a $[J] \sim \dfrac{[T]}{[X_R]}$

Mais $X_R \overset{\vee}{=} \dfrac{C_R° - C_R}{C_R°}$ n'a pas de dimension

donc $[J] \sim [T]$ $\underline{\text{J en K}}$

17) $\underline{J} = \dfrac{360 \times 10^3 \times 500}{10^3 \cdot 1800} = \underline{100 \text{ K}}$

18) Pour un fonctionnement adiabatique, il n'y a pas d'échange de chaleur avec l'extérieur, donc on élimine la conducto-convection : ainsi (4) devient

$$\boxed{\frac{dT}{dt} = J \frac{dX_R}{dt}}$$

(2)

19) J est une constante ; $dT = J dX_R$ s'intègre en $T - T_0 = J(X_R - X_R°)$

mais $X_R° = 0$ $\left(X_R = \dfrac{C_R° - C_R}{C_R°}\right)$

$$\boxed{T = T_0 + J X_R}$$

20) La fin de réaction correspondra à $X_R = 1$ (quand $C_R = 0$)

Alors $\boxed{T_f = T_0 + J} = 320 + 100 = \underline{420 \text{ K}}$

Or l'énoncé indique $T_{max} = 1{,}25 \, T_p = 400 \text{K}$

La marche adiabatique dépasse la t° maximale de sécurité d'où instabilité du réacteur dans la marche adiabatique.

J correspond à l'augmentation de température dans une marche adiabatique pour une réaction ayant tout consommé.

Le corrigé de la modélisation m'a été gracieusement apporté par un collègue.

21) Enoncé $y_i = \widehat{\beta}_1 \cdot x_i + \widehat{\beta}_0$

$\rightarrow \boxed{\frac{\partial y_i}{\partial \widehat{\beta}_0} = 1}$ et $\boxed{\frac{\partial y_i}{\partial \widehat{\beta}_1} = x_i}$

En notation Python : $\boxed{L[i,0] = 1 \mid L[i,1] = X[i]}$

22)
```
n = len(X)   # X a n éléments
L = np.zeros((n,2))   # L a n lignes 2 colonnes remplies de 0

for i in range(n):   # on modifie les éléments ligne par ligne
L[i,0] = 1
L[i,1] = X[i]
```

23) tL : transposée de L
```
tL = np.zeros((2,n))
for i in range(n) :
  tL [0,i] = L[i,0]
  tL [1,i] = L[i,1]
```

$P = L^t \times L$ est de dimension 2·2 car produit de 2 matrices de dimensions $(2,n)$ par $(n,2)$

```
P = np.zeros((2,2))
for i in range(2) :
    for j in range(2) :
        for k in range(n):
            P[i,j] = P[i,j] + tL[i,k] x L[k,j]
```
(ou aussi `P[i,j] = P[i,j] + L[k,i] x L[k,j]`)

24) $Q = L^t \cdot Y$ est de dimension $(2,1)$ car produit de 2 matrices de dimensions $(2,n)$ par $(n,1)$

```
Q = np.zeros((2,1))
for i in range(2)
   for k in range(n):
      Q[i,0] = Q[i,0] + tL[i,k] x Y[k]
```

25) Si $M = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ alors $M^{-1} = \frac{1}{\det M}\begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$

avec $\det M = ad - bc$

```
def inv-mat(M) :
    a, b, c, d = M[0,0], M[0,1], M[1,0], M[1,1]
                              # on nomme les coefficients
ref = np.array((2,2))   # on crée l'inverse remplie de 0
det = a*d - b*c         # calcul du déterminant
if det != 0 :
    ref [0,0] = d/det
    ref [0,1] = -b/det
    ref [1,0] = -c/det
    ref [1,1] = a/det
else
    return "erreur"
```

26) Enoncé : $\widehat{B} = (L^t \times L)^{-1} \times L^t Y$

en 23) P correspond à $L^t \times L$ donc on va inverser P :

```
inv P = inv-mat(P)
Bchap = np.array((2,1))   # B̂ est une matrice (2,1)=(2,2)x(2,1)
Bchap [0,0] = invP[0,0] x Q[0,0] + invP[0,1] x Q[1,0]
Bchap [1,0] = invP[1,0] x Q[0,0] + invP[1,1] x Q[1,0]
beta chap 0 = Bchap[0,0]   # donne β̂0
beta chap 1 = Bchap[1,0]   # donne β̂1
```

27) On connaît $X_R(t+\Delta t) = X_R(t) + \frac{dX}{dt}(t)\Delta t$

Si on fait le même principe dans le sens rétrograde, on aura $X_R(t) = X_R(t+\Delta t) - \frac{dX}{dt}(t+\Delta t)\Delta t$

d'où $\boxed{X_R(t+\Delta t) = X_R(t) + \Delta t \frac{dX}{dt}(t+\Delta t)}$

ce qui finalement ne change pas grand chose, que l'on prenne la tangente en t ou en $t+\Delta t$.

28) On en déduit : $\boxed{\frac{dX}{dt}(t+\Delta t) = \frac{X_R(t+\Delta t) - X_R(t)}{\Delta t}}$

29) De m $\boxed{T(t+\Delta t) = T(t) + \Delta t \frac{dT}{dt}(t+\Delta t)}$

30) $\boxed{\frac{dT}{dt}(t+\Delta t) = \frac{T(t+\Delta t) - T(t)}{\Delta t}}$

31) $$\boxed{\left.\frac{dX_R}{dt}\right|_{t_{i+1}} = \frac{X_{R_{i+1}} - X_{R_i}}{\Delta t}}$$

32) Enoncé (6) : $\frac{dX_R}{dt} = f_1$ et $\frac{dT}{dt} = f_2$

donc $$\boxed{X_{R_{i+1}} = X_{R_i} + \Delta t\, f_1(t_{i+1}, \text{etc})}$$

de m̂

33) $$\left.\frac{dT}{dt}\right|_{t_{i+1}} = \frac{T_{i+1} - T_i}{\Delta t}$$

34) $$T_{i+1} = T_i + \Delta t\, f_2(t_{i+1}, \text{etc})$$

35) On met tout du même côté !

$$0 = g_1(X_{R_{i+1}}, T_{i+1}) = X_{R_{i+1}} - X_{R_i} - \Delta t\, f_1(t_{i+1}, X_{R_{i+1}}, T_{i+1})$$

$$0 = g_2(X_{R_{i+1}}, T_{i+1}) = T_{i+1} - T_i - \Delta t\, f_2(t_{i+1}, X_{R_{i+1}}, T_{i+1})$$

Pour aller + loin il faut reprendre les résultats obtenus en 13 et 15

$$\frac{dX_R}{dt} = k_0 e^{-\frac{E_a}{RT}}(1 - X_R) = f_1$$

$$\frac{dT}{dt} = J k_0 e^{-\frac{E_a}{RT}}(1 - X_R) - \frac{T - T_p}{\tau_c}$$

d'où $$\boxed{\begin{aligned} g_1(X_{R_{i+1}}, T_{i+1}) &= X_{R_{i+1}} - X_{R_i} - \Delta t\, k_0 e^{-\frac{E_a}{RT_{i+1}}}(1 - X_{R_{i+1}}) = 0 \\ g_2(X_{R_{i+1}}, T_{i+1}) &= T_{i+1} - T_i - \Delta t\, J k_0 e^{-\frac{E_a}{RT_{i+1}}}(1 - X_{R_{i+1}}) - \frac{\Delta t}{\tau_c}(T_{i+1} - T_p) = 0 \end{aligned}}$$

36) On dérive : $$\boxed{\frac{\partial g_1}{\partial X_{R_{i+1}}} = 1 + k_0 \Delta t\, e^{-\frac{E_a}{RT_{i+1}}}}$$

$$\boxed{\begin{aligned} \frac{\partial g_1}{\partial T_{i+1}} &= -\Delta t\, k_0 \frac{E_a}{RT_{i+1}^2} e^{-\frac{E_a}{RT_{i+1}}}(1 - X_{R_{i+1}}) \\ \frac{\partial g_2}{\partial X_{R_{i+1}}} &= \Delta t\, J k_0 e^{-\frac{E_a}{RT_{i+1}}} \\ \frac{\partial g_2}{\partial T_{i+1}} &= 1 - J k_0 \Delta t \frac{E_a}{RT_{i+1}^2}(1 - X_{R_{i+1}}) - \frac{\Delta t}{\tau_c} \end{aligned}}$$

37) Enoncé : hyp $g_1T = \frac{\partial g_1}{\partial T_{i+1}}$ créée

ainsi que $g_1X = \frac{\partial g_1}{\partial X_{R_{i+1}}}$ $g_2T = \frac{\partial g_2}{\partial T_{i+1}}$ $g_2X = \frac{\partial g_2}{\partial X_{R_{i+1}}}$

et g_1 et g_2

```
def mat_Dg(x) :   # x = [T, X_R] à l'itération j = i+1
    Dg = np.zeros((2,2))
    Dg[0,0] = g1T(x[0], x[1])
    Dg[0,1] = g1X(x[0], x[1])
    Dg[1,0] = g2T(x[0], x[1])
    Dg[1,1] = g2X(x[0], x[1])
    return Dg
```

38)
```
invDg = inv_mat(mat_Dg(x_old))
T_new = x_old[0] - invDg[0,0]*g1(x_old[0], x_old[1])
        - invDg[0,1]*g2(x_old[0], x_old[1])

X_new = x_old[1] - invDg[1,0]*g1(x_old[0], x_old[1])
        - invDg[1,1]*g2(x_old[0], x_old[1])

x_new = [T_new, X_new]
```

39)
```
x_new = [Tp + J/2, 0.5]
x_old = [0, 0]
while abs(x_old[0] - x_new[0]) > 1E-5 :
    x_old = x_new
```

pas que la boucle démarre prendre x old & x new ≠

40)
```
m = int((tf - t0) / Deltat)
```

41)
```
t0 = 0
X0 = 0
T0 = Tp
t_euler = [t0]
X_euler = [X0]
T_euler = [T0]
for i in range(m)
    t_euler.append(t_euler[-1] + Deltat)
    x_new = [Tp, 0]
    x_old = [0, 0]
    for i in range(m) :
        while abs(. . .) > 1E-5 :
            x_old = x_new
        T_euler.append(x_new[0])
        X_euler.append(x_new[1])
```

42)
```
plt.plot(t_euler, T_euler)
plt.title("Evolution de la t° en fonction du temps")
plt.xlabel("temps (s)")
plt.ylabel("température (K)")
plt.show()
```