

## DEVOIR MAISON N° 7 : Réduction d'endomorphismes

L'usage d'une calculatrice était autorisé pour cette épreuve.

Un industriel cherche à optimiser son processus de cuisson des petits pois dans une cuve chauffée. Le cahier des charges demande à ce que les petits pois cuisent 10 minutes à une température comprise entre 90°C et 92°C. L'industriel utilise la puissance de chauffe pour commander les températures.

### 1 Modélisation et commande du processus

Dans cette partie, on suppose que la température des petits pois et la température de l'eau de la cuve sont uniformes et que la cuisson peut donc être modélisée par le système d'équations différentielles suivant :

$$\begin{cases} m_1 C_1 \frac{dT_1}{dt} = h_1 S_1 (T_2 - T_1) \\ m_2 C_2 \frac{dT_2}{dt} = h_1 S_1 (T_1 - T_2) + h_2 S_2 (T_\infty - T_2) + u \end{cases} \quad (E)$$

où  $m_1$  et  $m_2$  sont les masses respectives des petits pois et de l'eau dans la cuve (en kg),  $C_1$  et  $C_2$  sont les capacités thermiques massiques respectives du petit pois et de l'eau (en  $J \cdot K^{-1} \cdot kg^{-1}$ ),  $h_1$  et  $h_2$  sont les coefficients de transfert thermique entre l'eau et les petits pois d'une part, et entre l'eau et l'extérieur d'autre part (en  $W \cdot m^{-2} \cdot K^{-1}$ ),  $S_1$  et  $S_2$  sont respectivement la surface totale des petits pois et la surface d'échange entre l'eau et l'extérieur (en  $m^2$ ),  $T_1$ ,  $T_2$  et  $T_\infty$  sont les températures respectives des petits pois, de l'eau et de l'extérieur (en K) et  $u$  est la puissance de la résistance chauffante utilisée pour chauffer la cuve (en W). Cette puissance de chauffe  $u$  constitue le moyen d'action de l'industriel sur les températures  $T_1$  et  $T_2$  et est donc appelée la **commande**.

On suppose que toutes ces quantités sont constantes au cours du temps, en dehors des températures  $T_1$  et  $T_2$  et de la commande  $u$ . En particulier, la température extérieure est constante et vaut  $T_\infty = 20^\circ C$ . Le temps  $t$  sera exprimé en **minutes**.

#### 1.1 Commande en boucle ouvert (cas $u$ constant)

- ① On note  $x_1 = T_1 - T_\infty$  et  $x_2 = T_2 - T_\infty$ . Établir les équations différentielles vérifiées par  $x_1$  et  $x_2$ . On les écrira sous la forme :

$$\begin{cases} \frac{dx_1}{dt} = f_1(x_1, x_2, u) \\ \frac{dx_2}{dt} = f_2(x_1, x_2, u) \end{cases} .$$

On admettra que l'on peut réécrire ces équations sous la forme :

$$\frac{dx}{dt} = Ax + bu, \quad (1)$$

avec  $x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ ,  $\frac{dx}{dt} = \begin{pmatrix} \frac{dx_1}{dt} \\ \frac{dx_2}{dt} \end{pmatrix}$ ,  $A$  une matrice de  $\mathcal{M}_{2,2}(\mathbb{R})$ ,  $b$  un vecteur de  $\mathbb{R}^2$  et  $u \in \mathbb{R}$  constant.

**Dans la suite du problème, on prendra**

$$A = \begin{pmatrix} -5 & 5 \\ 4 & -6 \end{pmatrix} \text{ et } b = \begin{pmatrix} 0 \\ 9 \end{pmatrix} .$$

- ② On suppose qu'il existe  $x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$  dans  $\mathbb{R}^2$  et  $u$  dans  $\mathbb{R}$  tels que  $Ax + bu = 0$ .
- a. Montrer qu'alors  $x_1 = x_2$ .

- b. Interpréter.
  - c. Calculer la valeur de  $u$  si on considère que  $x_1 = x_2 = 72$ .
- ③
- a. Calculer les valeurs propres de  $A$ .
  - b. Montrer que  $A$  est diagonalisable.
  - c. Donner une base de vecteurs propres de  $A$ . On choisira les premières coordonnées de ces vecteurs propres égales à 1.
  - d. En déduire les matrices  $P$  et  $P^{-1}$  vérifiant  $A = PDP^{-1}$  avec  $D = \begin{pmatrix} -1 & 0 \\ 0 & -10 \end{pmatrix}$ .
- ④ Soit  $x$  une solution de (1). On définit  $z = P^{-1}x$ , où  $z = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix}$ .
- a. Montrer que  $\frac{dz}{dt} = P^{-1}\frac{dx}{dt}$ .
  - b. En déduire que

$$\frac{dz}{dt} = Dz + \beta u, \quad (2)$$

où  $\beta$  est un vecteur à expliciter.

- c. On suppose que  $T_1(0) = 20^\circ\text{C}$  et  $T_2(0) = 92^\circ\text{C}$ . Calculer  $z(0)$ .
- d. Résoudre l'équation différentielle vectorielle (2) en supposant que  $u = 16$ . On écrira séparément les équations différentielles vérifiées par  $z_1$  et  $z_2$ .
- e. En utilisant la relation  $x = Pz$ , montrer que pour tout  $t \geq 0$  :

$$\begin{cases} x_1(t) = 72 - 40e^{-t} - 32e^{-10t} \\ x_2(t) = 72 - 32e^{-t} + 32e^{-10t} \end{cases} .$$

- ⑤
- a. Étudier sur  $\mathbb{R}_+$  la fonction  $x_2$  définie à la question 4.(e). Tracer sa courbe représentative. *On fera figurer sur le graphique tous les éléments importants.*
  - b. Donner une interprétation du comportement de la fonction  $x_2$ .
  - c. On définit  $t_R = \min \{t \geq 0 ; \forall t' \geq t, T_2(t') \geq 90\}$ . Que représenter ce temps  $t_R$ ? Le placer sur la courbe représentative de  $x_2$ . On rappelle que  $x_2 = T_2 - T_\infty$ .
  - d. Montrer que  $\ln(16)$  minutes est une bonne approximation de  $t_R$ . Comment chaque valeur propre de  $A$  influence-t-elle  $t_R$ ?

## 1.2 Commande en boucle fermée (cas $u$ variable)

Pour améliorer les performances, on décide de faire varier la puissance de chauffe en fonction de la température de l'eau en considérant, pour tout  $t \geq 0$  :  $u(t) = 16 + k(72 - x_2(t))$ , où  $k \geq 0$  est une constante de réglage appelée **gain**.

- ① Expliquer ce choix de fonction pour la puissance de chauffage. Quel est l'intérêt?
- ② Que vaut  $t_R$  pour  $k = 0$ ?
- ③ On définit  $w(t) = x(t) - \begin{pmatrix} 72 \\ 72 \end{pmatrix}$ . Montrer que  $\frac{dw}{dt} = (A - kbc^\top)w$ , où  $c = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$  et  $c^\top$  désigne sa transposée.
- ④ On note  $\lambda_1(k)$  et  $\lambda_2(k)$  les valeurs propres de  $A - kbc^\top$  telles que  $\lambda_1(k) < \lambda_2(k)$ .
  - a. Donner l'expression de  $\lambda_1(k)$  et  $\lambda_2(k)$  en fonction de  $k$ .
  - b. Montrer que  $\lambda_1$  et  $\lambda_2$  sont des fonctions décroissantes de  $k$  sur  $\mathbb{R}_+$ .
- ⑤ Quel est l'effet du gain  $k$  sur  $t_R$ ? Quels inconvénients à augmenter  $k$  peut-on envisager?

## 2 Étude informatique des commandes

Cette partie consiste à mettre en place des programmes informatiques en lien avec la partie I.

Les programmes sont à rédiger en langage Python. L'annexe 1 page ?? comporte des rappels sur les commandes utiles. Avant chaque algorithme, on écrira brièvement le raisonnement suivi et la formule qu'il est censé calculer.

### 2.1 Recherche du minimum d'une fonction

Soit  $f : t \mapsto 72 - 32e^{-t} + 32e^{-10t}$ .

- ① Écrire une fonction d'en-tête `f(t)` prenant en entrée  $t$  (un réel) et qui renvoie la valeur de  $f(t)$ .
- ② On définit la fonction d'en-tête `minimum_f(N)` prenant en entrée  $N$  (un entier) par le code (incomplet) suivant :

```
1 def minimum_f(N):
2     Lt1 = np.linspace(0, 5, N)
3     Ly1 = []
4     for k in range(0, N):
5         Ly1.append(f(Lt1[k]))
6     ...
7     ...
8     return ...
```

Décrire en quelques mots ce que contiennent les variables `Lt1` et `Ly1` à l'issue des lignes 2 à 5.

- ③ Compléter sur la copie (avec autant de lignes que nécessaire) la fonction `minimum_f` pour qu'elle renvoie une liste composée des deux éléments suivants :
  - une valeur approchée  $m$  du minimum de  $f$  sur  $[0; 5]$  ;
  - le temps  $t_m$  en lequel cette valeur approchée est atteinte (c'est-à-dire  $m = f(t_m)$ ).

*On veillera à n'écrire qu'une seule boucle après celle des lignes 4 et 5 et à ne pas utiliser la commande `min`.*

- ④ On teste la fonction `minimum_f` sur différentes valeurs de  $N$ . Voici les résultats obtenus (tronqués à 4 chiffres après la virgule) :
  - pour  $N_1 = 20$ , la fonction donne  $m_1 = 49.7070$  et  $t_{m_1} = 0.2631$  ;
  - pour  $N_2 = 20\,000$ , la fonction donne  $m_2 = 49.7012$  et  $t_{m_2} = 0.2557$ .

Il y a une différence entre  $t_{m_1}$  et  $t_{m_2}$ .

- a. Cette différence était-elle prévisible et à quoi est-elle due ?
- b. Est-ce  $t_{m_1}$  ou  $t_{m_2}$  qui devrait être le plus proche de la valeur exacte cherchée ?

- ⑤ On teste maintenant sur une troisième valeur située entre  $N_1$  et  $N_2$  : pour  $N_3 = 50$ , la fonction donne  $m_3 = 49.9370$  et  $t_{m_3} = 0.3061$ . Commenter ce résultat.

### 2.2 Recherche du point d'annulation d'une fonction

Pour une fonction  $F : I \rightarrow \mathbb{R}$  donnée, avec  $I$  un intervalle de  $\mathbb{R}$ , on s'intéresse à la résolution approchée de l'équation  $F(x) = 0$ . On suppose que  $F$  s'annule en un point appelé  $x^* \in I$  et que  $F$  soit suffisamment régulière pour que les calculs suivants soient bien définis. L'objectif est d'obtenir une valeur approchée de  $x^*$ .

On utilise la méthode de Newton, qui consiste à définir la suite  $(u_n)_{n \in \mathbb{N}}$  par :

$$\begin{cases} u_0 \text{ à choisir} \\ u_{n+1} = u_n - \frac{F(u_n)}{F'(u_n)} \quad \text{pour } n \geq 0 \end{cases} .$$

On admet que, sous des hypothèses non précisées ici, cette suite converge vers  $x^*$  et que, s'il y a deux solutions à l'équation  $F(x) = 0$ , alors selon le choix de  $u_0$  la suite va converger vers une solution ou vers l'autre.

- ① Soit  $e > 0$ . À l'aide d'une boucle `while`, écrire une fonction d'en-tête `Newt(F, G, u0, e)` qui calcule les termes de la suite en s'arrêtant au premier terme  $u_{n_0}$  vérifiant  $|F(u_{n_0})| < e$  (en supposant qu'un tel  $n_0$  existe). Les arguments d'entrée sont :  $F$  une fonction,  $G$  une fonction qui correspond à  $F'$ ,  $u_0$  le premier terme de la suite et  $e$ . La fonction doit renvoyer  $u_{n_0}$ .
- ② Par rapport à l'objectif de résoudre de façon approchée l'équation  $F(x) = 0$ , expliquer à quoi correspond la quantité  $e$  utilisée dans l'écriture de `Newt`. Parmi les choix suivants pour  $e$ , indiquer (avec justification succincte) celui qui semble le plus pertinent pour obtenir la meilleure précision sur la valeur approchée :  $e_1 = 100$ ,  $e_2 = 1$ ,  $e_3 = 10^{-8}$ .
- ③ Toujours par rapport à l'objectif de résoudre de façon approchée l'équation  $F(x) = 0$ , justifier s'il est pertinent ou pas d'effectuer la modification suivante sur la fonction `Newt` : renvoyer le premier terme  $u_n$  vérifiant  $F(u_n) = 0$ .
- ④ À la question 6., on a supposé l'existence d'un  $n_0$  vérifiant  $|F(u_{n_0})| < e$ . Que se passe-t-il dans la fonction `Newt` si un tel  $n_0$  n'existe pas ?
- ⑤ On souhaite modifier la fonction `Newt` de manière à fixer un nombre maximal d'itérations pour la boucle `while`. Écrire une nouvelle fonction d'en-tête `NewtS(F, G, u0, e, nder)`, basée sur la fonction `Newt`, telle que :
  - s'il existe un terme  $u_{n_0}$  tel que  $|F(u_{n_0})| < e$  avec  $n_0 \leq n_{\text{der}}$ , la fonction renvoie  $u_{n_0}$  ;
  - si le nombre d'itérations dépasse (strictement) la valeur  $n_{\text{der}}$ , la boucle est stoppée et la fonction renvoie le booléen `False`.
- ⑥ Expliquer comment la méthode de Newton peut être utilisée pour trouver une valeur approchée du temps  $t_R$  défini à la question 5.(c) de la partie 1.
- ⑦ On définit les fonctions suivantes (où  $f$  a été définie à la partie 2.1) :

```
def F1(t):
    return f(t)-70
def G1(t) :
    return 32*np.exp(-t)-320*np.exp(-10*t)
```

puis on exécute pour  $e = 10^{-14}$  et  $nder = 9$  les deux appels suivants :

- `NewtS(F1, G1, 2, e, nder)` qui renvoie 2.7725887222252261 ;
- `NewtS(F1, G1, 0.1, e, nder)` qui renvoie 0.0072246696343453579.

Interpréter ces résultats en lien avec la partie 1.1.

« FIN DU SUJET »

---