CONCOURS BLANC — DEVOIR SURVEILLÉ # 4

le Mercredi 05/01/2022, 8h ⟶ 11h30

Épreuve de format « **Modélisation mathématique et informatique** » du concours A-ENV.

Consignes La qualité de la rédaction, la clarté et la précision des raisonnements, entreront pour une part importante dans l'appréciation de la copie. Les abréviations, sigles ou phrases nominales sont à proscrire. La numérotation des exercices (et des questions) doit être respectée et mise en évidence. Les résultats doivent être encadrés proprement.

Chaque candidat est responsable de la vérification de son sujet d'épreuve : pagination et impression de chaque page. Si, au cours de l'épreuve, un candidat repère ce qui lui semble être une erreur d'énoncé, il convient de le signaler sur la copie et de poursuivre la composition en expliquant les raisons des initiatives qui ont été prises.

L'usage de la calculatrice est interdit.

Exercice 1 ***Agro—Véto 2019. Modélisation de déplacements de fourmis, « ant algorithm »*** Pour se rendre d'un endroit à un autre, les individus d'une fourmilière ont le choix entre deux trajets disjoints, que nous nommerons A et B. À chaque fois qu'une fourmi emprunte l'un des deux chemins, elle y dépose une certaine quantité de phéromone qui peut éventuellement dépendre de la quantité de phéromone déjà présente sur le chemin.

Notations : pour chaque $n \geqslant 1$, α_n (respectivement β_n) désigne la quantité de phéromone présente sur le chemin A (*resp.* B) après le n^{e} trajet. A_n (*resp.* B_n) désigne l'évènement « la n^{e} fourmi choisit le trajet A (*resp.* B) ». Nous supposerons que chaque fourmi choisit de façon aléatoire le chemin qu'elle emprunte, en affectant à chacun une probabilité proportionnelle à la quantité de phéromone qui y est présente.

On a donc :

$$\mathbf{P}(A_{n+1} \mid [\alpha_n = a] \cap [\beta_n = b]) = \frac{a}{a+b}, \quad \mathbf{P}(B_{n+1} \mid [\alpha_n = a] \cap [\beta_n = b]) = \frac{b}{a+b}.$$

Enfin, X_n désignera le nombre de fourmis ayant choisi le trajet A lors des n premiers trajets.

Nous supposerons qu'initialement $\alpha_0 = \beta_0 = 1$ et qu'à chaque trajet une fourmi multiplie par un facteur $r > 1$ la quantité de phéromone déjà présente sur le chemin qu'elle emprunte.

1. Déterminer la loi des variables X_1, X_2.
2. Rédiger une fonction `simulX` qui reçoit un entier n et un réel r, simule les déplacements de n fourmis suivant la règle énoncée et renvoie le nombre X_n de fois où le chemin A a été emprunté.
3. Exprimer en fonction de n et de r la probabilité $\mathbf{P}(X_n = n)$. *On ne tentera pas de simplifier l'expression obtenue.*
4. On pose :

$$\forall n \geqslant 1, \quad p_n(r) = \frac{r}{1+r} \cdots \frac{r^n}{1+r^n},$$
$$\forall n \geqslant 0, \quad q_n(r) = \left(1 - \frac{1}{r}\right)\left(1 - \frac{1}{r^3}\right) \cdots \left(1 - \frac{1}{r^{2n+1}}\right).$$

   Démontrer que pour tout $r > 1$, les suites $(p_n(r))_{n \geq 1}$ et $(q_n(r))_{n \in \mathbf{N}}$ convergent.
   Indication : On commencera par étudier leur monotonie.
   On notera $p(r)$ et $q(r)$ leurs limites respectives.
5. En remarquant que :

$$\forall n \geqslant 1, \quad p_n(r) = \frac{1}{1+r^{-1}} \cdots \frac{1}{1+r^{-n}},$$

   montrer que :

$$p(r) \geqslant \exp\left(-\frac{1}{r-1}\right).$$

   Indication : On pourra admettre sans le démontrer l'inégalité suivante : $\forall x \in \mathbf{R}$, $e^x \geqslant 1 + x$..

6. Démontrer que :

$$\forall r > 1, \quad q(r) \leqslant \exp\left(-\frac{r}{r^2-1}\right).$$

7. On admet que : $\forall r > 1$, $p(r) = q(r)$. Déduire des questions précédentes un encadrement de la limite de la probabilité $\mathbf{P}(X_n = n)$ en fonction de r.
8. **(Conclusion :)** ce modèle vous semble-t-il approprié pour rendre compte du comportement des fourmis dans la réalité ?

Solution (exercice 1)

1. - Puisque $\alpha_0 = \beta_0 = 1$, la première fourmi choisit initialement chaque trajet avec probabilité $\frac{1}{2}$, donc $X_1(\Omega) = \{0,1\}$ et $\mathbf{P}(X_1 = 1) = \frac{1}{2}$ donc $\boxed{X_1 \hookrightarrow \mathscr{B}(1/2).}$
   - Pour X_2 : on a $X_2(\Omega) = \{0,1,2\}$. De plus,

$$\begin{aligned}
\mathbf{P}(X_2 = 0) &= \mathbf{P}\left({}^{c}A_1 \cap {}^{c}A_2\right) \\
&= \mathbf{P}({}^{c}A_2 | {}^{c}A_1)\,\mathbf{P}({}^{c}A_1) \\
&= \mathbf{P}({}^{c}A_2 | \alpha_1 = 1, \beta_1 = r\cdot 1)\cdot\frac{1}{2} \\
&= \frac{r}{1+r}\frac{1}{2} \\
&= \frac{r}{2(1+r)}
\end{aligned}$$

$$\begin{aligned}
\mathbf{P}(X_2 = 2) &= \mathbf{P}(A_1 \cap A_2) \\
&= \mathbf{P}(A_2 | A_1)\,\mathbf{P}(A_1) \\
&= \mathbf{P}(A_2 | \alpha_1 = r\cdot 1, \beta_1 = 1)\cdot\frac{1}{2} \\
&= \frac{r}{1+r}\frac{1}{2} \\
&= \frac{r}{2(1+r)}.
\end{aligned}$$

Donc $\mathbf{P}(X_1 = 1) = 1 - 2\frac{r}{2(1+r)} = \frac{r+1-r}{1+r} = \frac{1}{1+r}$. En conclusion :

$$\boxed{X_2(\Omega) = [\![0\,,2]\!], \quad \mathbf{P}(X_2 = 0) = \mathbf{P}(X_2 = 2) = \frac{r}{2(1+r)}, \mathbf{P}(X_2 = 1) = \frac{1}{1+r}.}$$

2. >_

```python
def simulX(n, r):
    X = 0
    a = 1
    b = 1
    for _ in range(n):
        if rd.random() < a/(a+b):
            # choix du trajet A
            X += 1
            a *= r
        else:
            # choix du trajet B
            b *= r
    return X
>>> simulX(100, 5)
1
>>> simulX(100, 1)
48
```

Pour la première exécution : le trajet choisit en premier est rapidement privilégié ; on observe donc des valeurs extrêmes pour X (proches de 100 ou proches de 0 en fonction du premier trajet choisi). Pour la seconde : l'équiprobabilité est maintenue, donc on observe des valeurs proches de $\frac{n}{2}$.

3. L'évènement $\{X_n = n\}$ signifie que toutes les fourmis ont choisi le même trajet A. Donc

$$\mathbf{P}(X_n = n) = \mathbf{P}(A_1 \cap \ldots A_n).$$

Donc par formule des probabilités composées :

$$\begin{aligned}
\mathbf{P}(X_n = n) &= \mathbf{P}(A_n | A_1 \cap \ldots A_{n-1}) \cdot \ldots \mathbf{P}(A_{n-1} | A_1 \cap \ldots A_{n-2}) \cdot \ldots \mathbf{P}(A_2 | A_1) \mathbf{P}(A_1) \\
&= \frac{r^{n-1}}{1+r^{n-1}} \cdot \cdots \cdot \frac{r}{1+r} \cdot \frac{1}{2} \\
&= \boxed{\prod_{i=0}^{n-1} \frac{r^i}{1+r^i}.}
\end{aligned}$$

Avec les notations introduites dans la suite, on a :

$$\boxed{\mathbf{P}(X_n = n) = p_{n-1}(r) \cdot \frac{1}{2}.}$$

4. On a des produits, des calculs de quotients sont donc plus adaptés pour la monotonie. Soit $n \in \mathbf{N}$, alors comme $p_n(r) \neq 0$ on a

$$\frac{p_{n+1}(r)}{p_n(r)} = \frac{r^{n+1}}{1+r^{n+1}} < 1.$$

La suite $(p_n(r))$ est donc décroissante, et minorée par zéro donc converge vers une limite finie $p(r)$. De-même :

$$\frac{q_{n+1}(r)}{q_n(r)} = 1 - \frac{1}{r^{2n+3}} < 1,$$

donc $(q_n(r))$ est décroissante, et minorée par zéro donc converge vers une limite finie $q(r)$.

$$\boxed{p_n(r) \xrightarrow{n\to\infty} p(r), \quad q_n(r) \xrightarrow{n\to\infty} q(r).}$$

5. Soit $n \geqslant 1$. En divisant les fractions au numérateur et dénominateur par $r, r^2, \ldots r^n$, on déduit :

$$\boxed{p_n(r) = \frac{1}{1+r^{-1}} \cdots \frac{1}{1+r^{-n}}},$$

donc en utilisant la majoration admise, dont la version inversée est :

$$\forall x > -1, \quad (1+x)^{-1} \geqslant e^{-x},$$

on déduit que

$$\begin{aligned}
p_n(r) &= \frac{1}{1+r^{-1}} \cdots \frac{1}{1+r^{-n}} \\
&= \prod_{k=1}^{n} (1+r^{-k})^{-1} \\
&\geqslant \prod_{k=1}^{n} e^{-r^{-k}} \\
&= e^{-\sum_{k=1}^{n} (r^{-1})^k} = e^{-r^{-1} \frac{1-(r^{-1})^n}{1-r^{-1}}} \\
&\geqslant e^{-\frac{1}{r-1}}.
\end{aligned}$$

somme géométrique et propriété de l'exp

$(r^{-1})^n \geqslant 0$

Donc en faisant $n \to \infty$: $\boxed{p(r) \geqslant \exp\left(-\frac{1}{r-1}\right).}$

6. On procède de la même manière.

$$\begin{aligned}
q_n(r) &= \prod_{k=0}^{n} \left(1 - \frac{1}{r^{2k+1}}\right) \\
&\leqslant \prod_{k=0}^{n} e^{-\frac{1}{r^{2k+1}}} \\
&= e^{-\frac{1}{r} \sum_{k=0}^{n} \left(\frac{1}{r^2}\right)^k} = e^{-\frac{1}{r} \frac{1-\left(\frac{1}{r^2}\right)^{n+1}}{1-\frac{1}{r^2}}} \\
&\geqslant e^{-\frac{1}{r} \frac{1}{1-\frac{1}{r^2}}} = e^{-\frac{r}{r^2-1}}.
\end{aligned}$$

majoration admise

somme géométrique et propriété de l'exp

$\left(\frac{1}{r^2}\right)^{n+1} \geqslant 0$

Donc en faisant $n \to \infty$: $\boxed{q(r) \leqslant \exp\left(-\frac{r}{r^2-1}\right).}$

7. On admet que $\forall r > 1$, $p(r) = q(r)$. On a déjà constaté que :

$$\forall n \geqslant 1, \quad \mathbf{P}(X_n = n) = p_{n-1}(r) \cdot \frac{1}{2}.$$

Donc la suite $(\mathbf{P}(X_n = n))$ converge vers $\frac{p(r)}{2}$. D'après les questions précédentes :

$$\boxed{\frac{1}{2}\exp\left(-\frac{1}{r-1}\right) \leqslant \lim_{n\to\infty} \mathbf{P}(X_n = n) \leqslant \frac{1}{2}\exp\left(-\frac{r}{r^2-1}\right).}$$

8. (Conclusion :) Lorsque r se rapproche de 1, on est censés retomber sur un contexte d'équiprobabilité, puisque les trajets déjà empruntés sont très peu renforcés en phéromone, donc la probabilité que sur n passages les fourmis choisissent toutes le même trajet devrait tendre vers zéro. Or, $\lim_{r\to1^+}\left(\frac{1}{2}\exp\left(-\frac{1}{r-1}\right)\right)=0$, $\lim_{r\to1^+}\left(\frac{1}{2}\exp\left(-\frac{r}{r^2-1}\right)\right)=0$, ce qui est donc cohérent avec l'intuition. Lorsque r devient grand, un trajet choisi est très rapidement renforcé en phéromone, et toutes les autres devraient suivre ce même trajet ensuite, avec forte probabilité : tout dépend donc du trajet emprunté par la toute première fourmi, A est choisi avec probabilité $\frac{1}{2}$. Mais,

$$\lim_{r\to\infty}\left(\frac{1}{2}\exp\left(-\frac{1}{r-1}\right)\right)=\frac{1}{2},\quad \lim_{r\to\infty}\left(\frac{1}{2}\exp\left(-\frac{r}{r^2-1}\right)\right)=\frac{1}{2}.$$

Là encore, c'est cohérent.

Problème 1 ***Contrôle d'un système de chauffe de petits pois – Modélisation 2020*** Le sujet se compose de 3 parties largement indépendantes. La partie II est consacrée à l'informatique. 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.

Partie I — 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 :

$$\left\{\begin{array}{lcl} m_1C_1\dfrac{dT_1}{dt} & = & h_1S_1(T_2-T_1) \\ m_2C_2\dfrac{dT_2}{dt} & = & h_1S_1(T_1-T_2)+h_2S_2(T_\infty-T_2)+u \end{array}\right. \tag{E}$$

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_∞ 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 $T_\infty = 20°C$. Le temps t sera exprimé en **minutes**.

I — 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 :

$$\left\{\begin{array}{lcl} \dfrac{dx_1}{dt} & = & f_1(x_1,x_2,u) \\ \dfrac{dx_2}{dt} & = & f_2(x_1,x_2,u) \end{array}\right..$$

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

$$\frac{dx}{dt} = Ax + bu, \tag{0.1}$$

avec $x=\begin{pmatrix} x_1\\ x_2\end{pmatrix}$, $\dfrac{dx}{dt}=\begin{pmatrix}\dfrac{dx_1}{dt}\\ \dfrac{dx_2}{dt}\end{pmatrix}$, A une matrice de $\mathfrak{M}_{2,2}(\mathbf{R})$, b un vecteur de $\mathbf{R}^2$ et $u\in\mathbf{R}$ 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 $\mathbf{R}^2$ et u dans $\mathbf{R}$ tels que $Ax + bu = 0$.

2.1) Montrer qu'alors $x_1 = x_2$.

2.2) Interpréter.

2.3) Calculer la valeur de u si on considère que $x_1 = x_2 = 72$.

3. **3.1)** Calculer les valeurs propres de A.

3.2) Montrer que A est diagonalisable.

3.3) Donner une base de vecteurs propres de A. On choisira les premières coordonnées de ces vecteurs propres égales à 1.

3.4) 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 $z = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix}$.

4.1) Montrer que $\dfrac{dz}{dt} = P^{-1}\dfrac{dx}{dt}$.

4.2) En déduire que

$$\frac{dz}{dt} = Dz + \beta u, \tag{0.2}$$

où β est un vecteur à expliciter.

4.3) On suppose que $T_1(0) = 20^\circ C$ et $T_2(0) = 92^\circ C$.Calculer $z(0)$.

4.4) Résoudre l'équation différentielle vectorielle (2) en supposant que $u = 16$. On écrira séparément les équations différentielles vérifiées par z_1 et z_2.

4.5) En utilisant la relation $x = Pz$, montrer que pour tout $t \geqslant 0$:

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

5. **5.1)** Étudier sur $\mathbf{R}_+$ la fonction x_2 définie à la question 4.(5).Tracer sa courbe représentative.*On fera figurer sur le graphique tous les éléments importants.*

5.2) Donner une interprétation du comportement de la fonction x_2.

5.3) On définit $t_R = \min\{t \geqslant 0\,;\, \forall t' \geqslant t,\ T_2(t') \geqslant 90\}$.Que représente ce temps t_R ? Le placer sur la courbe représentative de x_2.On rappelle que $x_2 = T_2 - T_\infty$.

5.4) Montrer que ln(16) minutes est une bonne approximation de t_R.Comment chaque valeur propre de A influence-t-elle t_R ?

I — 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 $t \geqslant 0$: $\quad u(t) = 16 + k\left(72 - x_2(t)\right)$, où $k \geqslant 0$ est une constante de réglage appelée **gain**.

6. Expliquer ce choix de fonction pour la puissance de chauffage. Quel est l'intérêt ?
7. Que vaut t_R pour $k = 0$?
8. On définit $w(t) = x(t) - \begin{pmatrix} 72 \\ 72 \end{pmatrix}$ pour tout $t \geqslant 0$. Montrer que $\dfrac{dw}{dt} = \left(A - kb^\top c\right)w$,

   où $c = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ et $^\top c$ désigne sa transposée.
9. On note $\lambda_1(k)$ et $\lambda_2(k)$ les valeurs propres de $A - kb^\top c$ telles que $\lambda_1(k) < \lambda_2(k)$.

   9.1) Donner l'expression de $\lambda_1(k)$ et $\lambda_2(k)$ en fonction de k.

   9.2) Montrer que λ_1 et λ_2 sont des fonctions décroissantes de k sur $\mathbf{R}_+$.
10. Quel est l'effet du gain k sur t_R ? Quels inconvénients à augmenter k peut-on envisager ?

Partie II — 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 comporte des rappels sur les commandes utiles. Avant chaque algorithme, on écrira brièvement le raisonnement suivi et la formule qu'il est censé calculer.

II — 1) Recherche du minimum d'une fonction

Soit $f : t \in \mathbf{R} \longmapsto 72 - 32e^{-t} + 32e^{-10t}$.

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 :

```python
def minimum_f(N):
    Lt1 = np.linspace(0, 5, N)
    Ly1 = []
    for k in range(0, N):
        Ly1.append(f(Lt1[k]))
    ...
    ...
    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}.

4.1) Cette différence était-elle prévisible et à quoi est-elle due ?

4.2) 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 $t_{m_2} = 0.3061$. Commenter ce résultat.

II — 2) Recherche du point d'annulation d'une fonction Pour une fonction $F : I \to \mathbf{R}$ donnée, avec I un intervalle de $\mathbf{R}$, 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 $(u_n)_{n\in\mathbf{N}}$ par :

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

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.

6. 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}.

7. 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 : $e_1 = 100$, $e_2 = 1$, $e_3 = 10^{-8}$.

8. 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 $F(u_n) = 0$.

9. 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 ?

10. 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 $n_0 \leqslant n_{\text{der}}$, la fonction renvoie u_{n_0} ;
- si le nombre d'itérations dépasse (strictement) la valeur n_{der}, la boucle est stoppée et la fonction renvoie le booléen `False`.

11. 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 I.

12. On définit les fonctions suivantes (où `f` a été définie à la partie II.1) :

```python
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 I.1.

Solution (problème 1)

J'adopte la même convention que le sujet original, à savoir que les éléments de $\mathbf{R}^2$ sont notés en colonne.

Partie I — Modélisation et commande du processus Rappelons le système différentiel satisfait par (T_1, T_2) :

$$\left\{\begin{array}{lcl} m_1C_1\dfrac{dT_1}{dt} & = & h_1S_1(T_2-T_1) \\ m_2C_2\dfrac{dT_2}{dt} & = & h_1S_1(T_1-T_2)+h_2S_2(T_\infty-T_2)+u \end{array}\right. \tag{E}$$

I — 1) Commande en boucle ouvert (cas u constant)

1. On note $x_1 = T_1 - T_\infty$ et $x_2 = T_2 - T_\infty$. Alors

$$\begin{aligned} m_1C_1\frac{dx_1}{dt} &= m_1C_1\frac{dT_1}{dt} \\ &= h_1S_1(T_2-T_1) \quad \text{\textit{en utilisant} (E)} \\ &= h_1S_1(T_2-T_\infty+T_\infty-T_1) \\ &= h_1S_1(x_2-x_1). \end{aligned}$$

De-même, nous obtenons pour x_2 :

$$\begin{aligned} m_2C_2\frac{dx_2}{dt} &= m_2C_2\frac{dT_2}{dt} \\ &= h_1S_1(T_1-T_2)-h_2S_2x_2+u \quad \text{\textit{en utilisant} (E)} \\ &= h_1S_1(x_1-x_2)-h_2S_2x_2+u. \end{aligned}$$

En résumé, nous avons obtenu :

$$\boxed{\begin{aligned} \frac{dx_1}{dt} &= \frac{h_1S_1}{m_1C_1}(x_2-x_1) \underset{\textbf{(défi.)}}{=} f_1(x_1,x_2,u), \\ \frac{dx_2}{dt} &= \frac{h_1S_1}{m_2C_2}(x_1-x_2)-\frac{h_2S_2}{m_2C_2}x_2+\frac{1}{m_2C_2}u \underset{\textbf{(défi.)}}{=} f_2(x_1,x_2,u). \end{aligned}}$$

On obtient alors que $x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ vérifie le système différentiel suivant (la dérivée étant comprise ligne par ligne) :

$$\frac{dx}{dt} = Ax + bu, \tag{0.3}$$

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

2. **2.1)** On résout simplement le système associé. En effet, x vérifie $Ax + bu = 0$ si, et seulement si :

$$\left\{\begin{array}{ll} -5x_1+5x_2+0 & = 0, \\ 4x_1-6x_2+9u & = 0 \end{array}\right. \iff \left\{\begin{array}{ll} x_1 & = x_2, \\ 2x_1 & = 9u \end{array}\right. .$$

En particulier, $\boxed{x_1 = x_2.}$

2.2) On peut proposer une interprétation algébrique, en l'occurence,

l'ensemble des solutions $x \in \mathbf{R}^2$ du système avec second membre $Ax = -bu$ est inclus dans $\operatorname{Vect}\begin{pmatrix} 1 \\ 1 \end{pmatrix}$.

Par rapport aux solutions $t \in \mathbf{R}^+ \longmapsto x(t) = (x_1(t), x_2(t))$, on peut dire que s'il existe $t_0 \in \mathbf{R}^+$ tel que $\dfrac{dx}{dt}(t_0) = 0$, alors $x_1(t_0) = x_2(t_0)$. Donc

$\boxed{\text{les temps d'annulation de la dérivée de } x \text{ ne peuvent être que des temps où la température de l'eau est égale à celle des petits poids.}}$[1]

2.3) Calculons la valeur de u si on considère que $x_1 = x_2 = 72$. D'après la seconde ligne du système, si $x_1 = x_2 = 72$, alors $u = \frac{2}{9}x_1 = \frac{2}{9}72 \boxed{= 16.}$

3. 3.1) Soit $\lambda \in \mathbf{C}$. Alors $\det(A - \lambda I_2) = \begin{vmatrix} -5-\lambda & 5 \\ 4 & -6-\lambda \end{vmatrix} = \lambda^2 + 11\lambda + 10$. Le discriminant associé est alors $\Delta = 81 = 9^2$, donc les deux racines sont

$$\frac{-11+9}{2} = -1 \quad \text{ou} \quad \frac{-11-9}{2} = -10.$$

Donc $\boxed{\mathrm{Spec}(A) = \{-10, -1\}.}$

3.2) La matrice A est de format 2×2, et possède deux valeurs propres distinctes, donc $\boxed{\text{est diagonalisable (sur } \mathbf{R}).}$

3.3) Commençons par calculer les éléments de $E_{-10}(A)$. Soit $X = \begin{pmatrix} x \\ y \end{pmatrix} \in \mathfrak{M}_{2,1}(\mathbf{R})$,

$$\begin{aligned} &(A + 10I_2)X = 0 \\ \iff &\begin{cases} 5x + 5y &= 0 \\ 4x + 4y &= 0 \end{cases} \\ \iff &x = -y. \end{aligned}$$

Donc $\boxed{E_{-10}(A) = \mathrm{Vect}\begin{pmatrix} 1 \\ -1 \end{pmatrix}.}$ Faisons de-même pour l'autre. Soit $X = \begin{pmatrix} x \\ y \end{pmatrix} \in \mathfrak{M}_{2,1}(\mathbf{R})$,

$$\begin{aligned} &(A + I_2)X = 0 \\ \iff &\begin{cases} -4x + 5y &= 0 \\ 4x - 5y &= 0 \end{cases} \\ \iff &4x = 5y. \end{aligned}$$

Donc $\boxed{E_{-1}(A) = \mathrm{Vect}\begin{pmatrix} 1 \\ \frac{4}{5} \end{pmatrix}.}$

3.4) Notons $\boxed{P = \begin{pmatrix} 1 & 1 \\ -1 & \frac{4}{5} \end{pmatrix}}$, alors P est inversible d'inverse $\boxed{P^{-1} = \frac{5}{9}\begin{pmatrix} \frac{4}{5} & -1 \\ 1 & 1 \end{pmatrix}.}$ D'après la formule de changement de base, et puisque A est diagonalisable, nous avons

$$\boxed{A = PDP^{-1}, \quad \text{avec} \quad D = \mathrm{Diag}(-10, -1).}$$

4. Soit x une solution de (1). On définit $z = P^{-1}x$, où $z = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix}$.

4.1) Notons $z = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix}$. Alors $P^{-1}x = \frac{5}{9}\begin{pmatrix} \frac{4}{5} & -1 \\ 1 & 1 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \frac{5}{9}\begin{pmatrix} \frac{4}{5}x_1 - x_2 \\ x_1 + x_2 \end{pmatrix}$.
Donc, d'après les règles d'opérations sur les dérivées des fonctions à valeurs

[1] avec Cauchy-Lipschitz, on pourrait dire beaucoup plus....

réelles, nous avons :

$$\frac{\mathrm{d}z}{\mathrm{d}t} = \frac{5}{9}\begin{pmatrix} \frac{4}{5}x_1' - x_2' \\ x_1' + x_2' \end{pmatrix} = \mathrm{P}^{-1}\begin{pmatrix} x_1' \\ x_2' \end{pmatrix} = \boxed{\mathrm{P}^{-1}\frac{\mathrm{d}x}{\mathrm{d}t}}.$$

4.2) Nous obtenons alors :

$$\begin{aligned}\frac{\mathrm{d}z}{\mathrm{d}t} &= \mathrm{P}^{-1}\frac{\mathrm{d}x}{\mathrm{d}t} \\ &= \mathrm{P}^{-1}(\mathrm{PDP}^{-1}x + bu) && \textit{ED vérifiée par } x \\ &= \mathrm{DP}^{-1}x + \mathrm{P}^{-1}bu \\ &= \boxed{\mathrm{D}z + \beta u, \quad \text{en posant } \beta = \mathrm{P}^{-1}b.} && \textit{définition de } z\end{aligned}$$

4.3) On suppose que $\mathrm{T}_1(0) = 20^\circ\mathrm{C}$ et $\mathrm{T}_2(0) = 92^\circ\mathrm{C}$. Alors :

$$z(0) = \mathrm{P}^{-1}x(0) = \frac{5}{9}\begin{pmatrix} \frac{4}{5} & -1 \\ 1 & 1 \end{pmatrix}\begin{pmatrix} 0 \\ 72 \end{pmatrix} = \frac{5}{9}\begin{pmatrix} -72 \\ 72 \end{pmatrix} = \boxed{40\begin{pmatrix} -1 \\ 1 \end{pmatrix}.}$$

4.4) Calculons pour commencer le vecteur β.

$$\beta = \frac{5}{9}\begin{pmatrix} \frac{4}{5} & -1 \\ 1 & 1 \end{pmatrix}\begin{pmatrix} 0 \\ 9 \end{pmatrix} = 5\begin{pmatrix} \frac{4}{5} & -1 \\ 1 & 1 \end{pmatrix}\begin{pmatrix} 0 \\ 1 \end{pmatrix} = 5\begin{pmatrix} -1 \\ 1 \end{pmatrix}.$$

L'équation (2) est équivalente à la suivante[2] :

$$\begin{cases} z_1'(t) &= -10z_1(t) - 5.16, \\ z_2'(t) &= -z_2(t) + 5.16 \end{cases}$$

$$\begin{cases} z_1(t) &= \mathrm{K}e^{-10t} - 8, \\ z_2(t) &= \mathrm{K}'e^{-t} + 5 \times 16 \end{cases}, \quad \mathrm{K}, \mathrm{K}' \in \mathbf{R}.$$

[2]version « décorrélée » du système initial

En tenant compte des conditions initiales $z_1(0) = -40$, $z_2(0) = 40$, on obtient comme conditions

$$\mathrm{K} - 8 = -40, \quad \mathrm{K}' + 5.16 = 40,$$

d'où finalement, pour tout $t \in \mathbf{R}^+$:

$$\boxed{z_1(t) = -32e^{-10t} - 8, \quad z_2(t) = -40e^{-t} + 5.16.}$$

4.5) En utilisant la relation $x = \mathrm{P}z$, on obtient pour tout $t \in \mathbf{R}^+$:

$$\begin{cases} x_1(t) &= z_1(t) + z_2(t) \\ x_2(t) &= -z_1(t) + \frac{4}{5}z_2(t). \end{cases}$$

Après calculs, on trouve le système de l'énoncé :

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

5. 5.1) Soit $t \in \mathbf{R}^+$. Alors la fonction x_2 est dérivable sur $\mathbf{R}^+$ en tant que somme de telles fonctions. Et,

$$x_2'(t) = 32e^{-10t} - 320e^{-10t}.$$

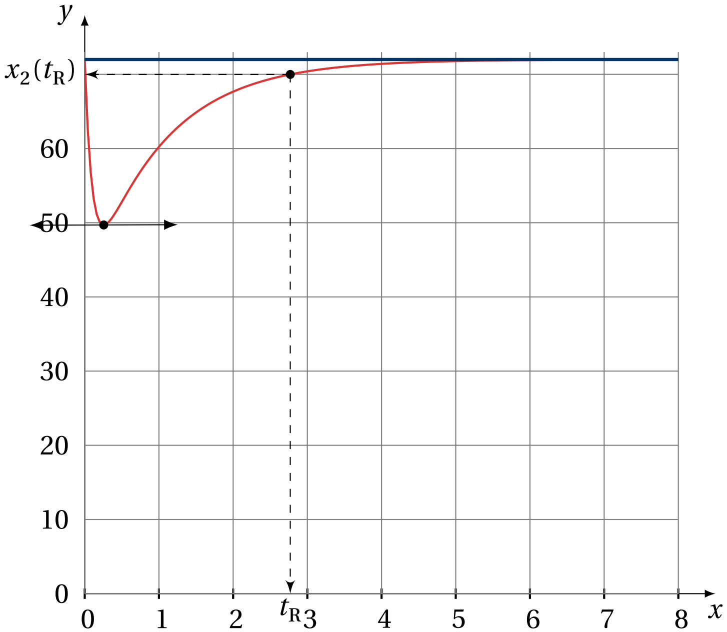
Dès lors, nous avons

$$\begin{aligned} x_2'(t) \geqslant 0 &\iff 32e^{-t} \geqslant 320e^{-10t} \\ &\iff -t \geqslant \ln 10 - 10t \\ &\iff 9t \geqslant \ln 10 \\ &\iff t \geqslant \frac{\ln 10}{9} (\geqslant 0). \end{aligned}$$

On déduit alors le tableau de variations ci-dessous, puis la courbe représentative de la fonction.

t	0	$\frac{\ln 10}{9}$	$+\infty$
$x_2(t)$	72 ↘	$x_2\left(\frac{\ln 10}{9}\right)$	↗ 72

La dérivée s'annulant en $\frac{\ln 10}{9}$, nous avons donc une pente horizontale en ce point.



5.2) Le processus commence avec une eau très chaude (92°C), on démarre ensuite la chauffe qui au début ne parvient pas à compenser la chute de température due à la perdition d'énergie. Dans un troisième temps, la température de l'eau augmente, conséquence du démarrage de la chauffe.

5.3)

$$t_R = \min\{t \geqslant 0\,;\ \forall t' \geqslant t,\ T_2(t') \geqslant 90\} = \min\{t \geqslant 0\,;\ \forall t' \geqslant t,\ x_2(t') \geqslant 70\}.$$

C'est donc le plus petit temps à partir duquel $\boxed{\text{la courbe de } x_2 \text{ reste après au-dessus de la droite } x = 70.}$ Ainsi, après ce temps, la courbe de chauffe x_2 respecte le cahier des charges imposé par l'industriel.

5.4) Nous avons $x_2(\ln(16)) = 72 - \frac{32}{16} + \frac{32}{16^{10}} = 70 + \frac{32}{16^{10}} \approx \boxed{70}$.

Donc $\boxed{\ln(16) \text{ est une bonne approximation de } t_R.}$ Les valeurs propres sont les coefficients dans les exponentielles décroissantes. $\boxed{\text{Plus les valeurs propres seront négatives, plus } x_2 \text{ convergera vite vers 72, donc plus } t_R \text{ sera petit.}}$

I — 2) Commande en boucle fermée (cas u variable)

6. Avec ce choix de chauffage, lorsque x_2 baisse alors u augmente. Ainsi, $\boxed{\text{ce système aura tendance à réguler la température de l'eau}}$ en l'augmentant/diminuant lorsque cela est nécessaire.

7. Si $k = 0$, alors u est constant et on tombe dans le cadre des questions précédentes. Ainsi : $\boxed{t_R \approx \ln(16)}$.

8. On définit $w(t) = x(t) - \begin{pmatrix} 72 \\ 72 \end{pmatrix}$ pour tout $t \geqslant 0$. Constatons que ${}^{\top}c.w = \begin{pmatrix} 0 & 1 \end{pmatrix}\begin{pmatrix} x_1 - 72 \\ x_2 - 72 \end{pmatrix} = x_2 - 72$. On en déduit alors la dérivée de w,

$$\begin{aligned}
\frac{\mathrm{d}w}{\mathrm{d}t} &= \frac{\mathrm{d}x}{\mathrm{d}t} \\
&= Ax + bu \\
&= A\left(w + 72\begin{pmatrix} 1 \\ 1 \end{pmatrix}\right) + b(16 + k(72 - x_2)) \\
&= A\left(w + 72\begin{pmatrix} 1 \\ 1 \end{pmatrix}\right) + b(16 - k\,{}^{\top}c.w) \\
&= (A - kb\,{}^{\top}c)w + 72A\begin{pmatrix} 1 \\ 1 \end{pmatrix} + 16b
\end{aligned}$$

Après calculs, on trouve que $72\mathrm{A}\begin{pmatrix}1\\1\end{pmatrix}+16b=\begin{pmatrix}0\\0\end{pmatrix}$, d'où la formule de l'énoncé :

$$\boxed{\frac{\mathrm{d}w}{\mathrm{d}t}=(\mathrm{A}-kb^\top c)w}.$$

9. On note $\lambda_1(k)$ et $\lambda_2(k)$ les valeurs propres de $\mathrm{A}-kb^\top c$ telles que $\lambda_1(k)<\lambda_2(k)$.

9.1) Calculons les deux valeurs propres à l'aide du déterminant. Nous avons

$$\mathrm{A}-kb^\top c=\begin{pmatrix}-5&5\\4&-6\end{pmatrix}-k\begin{pmatrix}0\\9\end{pmatrix}\begin{pmatrix}0&1\end{pmatrix}=\begin{pmatrix}-5&5\\4&-6\end{pmatrix}-k\begin{pmatrix}0&0\\0&9\end{pmatrix}.$$

Soit $\lambda\in\mathbf{C}$, alors calculons

$$\mathrm{D}(\lambda)\underset{\textbf{(défi.)}}{=}\begin{vmatrix}-5-\lambda&5\\4&-6-9k-\lambda\end{vmatrix}=(-5-\lambda)(-6-9k-\lambda)-20.$$

En développant, on obtient :

$$\mathrm{D}(\lambda)=\lambda^2+\lambda(9k+11)+5(9k+2).$$

Le discriminant de ce polynôme en λ est alors $(9k+11)^2-20(9k+2)=9(9k^2+2k+9)\geqslant 0$ puisque $k\geqslant 0$. Ainsi,

$$\boxed{\begin{aligned}&\mathrm{Spec}(\mathrm{A}-kb^\top c)\\&=\left\{\frac{-9k-11-3\sqrt{9k^2+2k+9}}{2},\frac{-9k-11+3\sqrt{9k^2+2k+9}}{2}\right\}.\end{aligned}}$$

On note alors $\lambda_1(k)=\dfrac{-9k-11-3\sqrt{9k^2+2k+9}}{2}$ et $\lambda_2(k)=\dfrac{-9k-11+3\sqrt{9k^2+2k+9}}{2}$.

9.2) La fonction λ_1 est une somme de fonctions décroissantes, donc $\boxed{\lambda_1\text{ est décroissante.}}$ Pour λ_2, nous voilà condamnés à faire les calculs à la main en calculant la dérivée. Soit $k\in\mathbf{R}^+$,

$$\begin{aligned}\lambda_2'(k)&=\frac{1}{2}\left(-9+\frac{3}{2}\frac{18k+2}{\sqrt{9k^2+2k+9}}\right)\\&=\frac{1}{2}\left(-9+3\frac{9k+1}{\sqrt{9k^2+2k+9}}\right)\\&=\frac{1}{2}\frac{-9\sqrt{9k^2+2k+9}+3(9k+1)}{2\sqrt{9k^2+2k+9}}\end{aligned}$$

Ainsi, on a

$$\begin{aligned}&\lambda_2'(k)\leqslant 0\\\iff\quad&3(9k+1)\leqslant 9\sqrt{9k^2+2k+9}\\\iff\quad&(9k+1)^2\leqslant 3^2(9k^2+2k+9)\\\iff\quad&81k^2+18k+1\leqslant 81k^2+18k+81\\\iff\quad&1\leqslant 81.\end{aligned}$$

quantités positives car $k\geqslant 0$

Ainsi la fonction λ_2' est négative sur $\mathbf{R}^+$, d'où $\boxed{\text{la décroissance de }\lambda_2\text{ sur }\mathbf{R}^+}$.

10. Plus k, plus les valeurs propres de $\mathrm{A}-kb^\top c$ seront petites dans les négatifs, donc plus le temps t_R sera petit d'après la question 5.4. En revanche, si l'on augmente trop la valeur de k, on augmente l'énergie à fournir pour la chauffe.

Partie II — 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 comporte des rappels sur les commandes utiles. Avant chaque algorithme, on écrira brièvement le raisonnement suivi et la formule qu'il est censé calculer.

II — 1) Recherche du minimum d'une fonction

Soit $f:t\in\mathbf{R}\longmapsto 72-32e^{-t}+32e^{-10t}$.

1.

```python
import numpy as np
def f(t):
    return 72 - 32*np.exp(-t) + 32*np.exp(-10*t)
```

Une déclaration alternative serait simplement

```python
f = lambda t:72 - np.exp(-t) + 32*np.exp(-10*t)
```

2. Après les lignes 2 et 5, la variable `Lt1` contient les éléments d'une subdivision régulière en N points de l'intervalle $[0,5]$, *i.e.* les $\boxed{\frac{5k}{N} \text{ avec } k \in [\![0\,,N]\!]}$. La variable `Ly1` contient les images par f de cette subdivision sauf le dernier, donc les $\boxed{f\left(\frac{5k}{N}\right) \text{ avec } k \in [\![0\,,N-1]\!]}$.

3. Il suffit alors d'effectuer une recherche de minimum sur les éléments de la liste `Ly1`.

```python
def minimum_f(N):
    Lt1 = np.linspace(0, 5, N)
    Ly1 = []
    for k in range(0, N):
        Ly1.append(f(Lt1[k]))
    # Recherche de minimum
    mini = Ly1[0]
    ind_mini = 0
    for i in range(1, N):
        if Ly1[i] < mini:
            mini = Ly1[i]
            ind_mini = i
    return [mini, Lt1[ind_mini]]
```

Plus précisément, la deuxième coordonnée du résultat sera la dernière valeur où le minimum est atteint. Testons par exemple avec un jeu de valeurs donné par l'énoncé : pour $N_1 = 20$, `minimum_f(20)` retourne bien

[49.70701495513559, 0.2631578947368421].

4. **4.1)** N_2 est bien supérieur à N_1. L'algorithme de calcul d'une valeur approchée du minimum risque d'être plus précis en utilisant N_2. La différence provient donc d'une différence dans les précisions choisies (dans le calcul de la subdivision de $[0,5]$).

4.2) Logiquement (mais encore faudrait-il le montrer), la valeur t_{m_2} associée à N_2 devrait être meilleure que t_{m_1}.

5. L'algorithme cherche à approcher $t^* = \dfrac{\ln 10}{9} \approx 0.2558$. Nous avons obtenu $m_2 \leqslant m_1 \leqslant m_3$ et $t_{m_2} \leqslant t^* \leqslant t_{m_1} \leqslant t_{m_3}$. Il n'est donc pas clair que la suite des erreurs commises $\left(\left|t_{m_N} - t^*\right|\right)_N$ est décroissante en N (la valeur obtenue pour N = 50 semble moins précise que celle pour N = 20), néanmoins cette suite semble converger vers zéro lorsque $N \longrightarrow \infty$ d'après le résultat de l'appel associé à N = 20000.

II — 2) Recherche du point d'annulation d'une fonction

Remarque : dans le cas où il existe deux solutions, la suite (u_n) n'est pas bien définie pour certains u_0

6. On construit les termes de la suite récurrente jusqu'à ce que la condition de terminaison $\left|F(u_n)\right| < e$ soit satisfaite.

```python
def Newt(F, G, u0, e):
    """
    retourne le terme de la suite de Newton à précision e
    """
    u = u0
    while np.abs(F(u)) >= e:
        u = u - F(u)/G(u)
    return u
```

Par exemple, si l'on considère $F(x) = x^2 - 2$ et $G(x) = 2x$ pour $x \in I = \mathbf{R}^+$, alors l'algorithme devrait retourner une valeur approchée de $\sqrt{2}$.

```python
def F(x):
    return x**2 - 2
def G(x):
    return 2*x
```

L'appel `Newt(F, G, 1, 10**(-4))` retourne 1.4142156862745099. C'est une bonne approximation de $\sqrt{2}$.

7. Tester l'égalité à zéro d'un flottant n'est absolument pas pertinent. On remplace donc cette condition d'arrêt par une majoration par une quantité très petite. Plus celle-ci est petite, meilleure sera l'approximation de x vérifiant $\mathrm{F}(x) = 0$. $\boxed{\text{On choisira donc plutôt } e_3 = 10^{-8}}$.

8. Cela n'est pas pertinent, toujours du fait de la gestion des flottants par Python puisque des valeurs approchées sont utilisées à chaque étape de l'algorithme.

9. Si un tel n_0 n'existe pas, alors la structure `while` ne s'arrêtera pas. On pourrait donc ajouter une borne sur le nombre d'itérations, c'est ce qui est fait dans la suite.

10. On souhaite modifier la fonction `Newt` de manière à fixer un nombre maximal d'itérations pour la boucle `while`. Il suffit alors d'ajouter une seconde condition d'arrêt.

```python
def NewtS(F, G, u0, e, nder):
    """
    retourne le terme de la suite de Newton à précision e avec
    borne sur le nombre d'itérations
    """
    u = u0
    n = 0 # on ajoute un compteur d'itérations
    while np.abs(F(u)) >= e and n <= nder:
        n += 1
        u = u - F(u)/G(u)
    if n <= nder:
        return u
    else:
        return False
```

Reprenons l'exemple de $\sqrt{2}$ mentionné ci-dessus. L'appel `NewtS(F, G, 1, 10**(-4), 10)` retourne 1.4142156862745099.

11. D'après le tableau de variations établi pour x_2, le temps t_R peut être définie comme le temps $t \geqslant t^* = \dfrac{\ln 10}{9}$ pour lequel $x_2(t) = 70$. Notons $\mathrm{F} \left| \begin{array}{ccc} [t^\star, \infty] & \longrightarrow & \mathbf{R} \\ t & \longmapsto & x_2(t) - 70 \end{array} \right.$.

$\boxed{\text{La méthode de Newton appliquée à la fonction F permettrait d'obtenir une valeur approchée de } t_\mathrm{R}.}$

Il faudrait cependant veiller à l'initialisation pour garantir la convergence vers t_R et pas vers l'autre solution.

12. Constatons tout d'abord que pour tout $t \in \mathbf{R}^+$,

$$\frac{\mathrm{dF}_1}{\mathrm{d}t}(t) = f'(t) = \mathrm{G}_1(t),$$

où $\mathrm{F}_1, \mathrm{G}_1$ sont les deux fonctions mathématiques associées aux fonctions Python `F_1` et `G_1`.

$\boxed{\text{Les appels retournent donc, sous réserve de convergence, une valeur approchée d'une solution de l'équation } x_2(t) = 70.}$

Pour le premier appel, vu le résultat, on constate que c'est une valeur approchée de t_R qui est retournée. En revanche, pour le second, c'est la solution sur l'intervalle $\left[0, \dfrac{\ln 10}{9}\right]$. La différence observée est due à l'initialisation.

CORRECTION

Solution (exercice 1)

1. ▸ Puisque $\alpha_0 = \beta_0 = 1$, la première fourmi choisit initialement chaque trajet avec probabilité $\frac{1}{2}$, donc $X_1(\Omega) = \{0,1\}$ et $\mathbf{P}(X_1 = 1) = \frac{1}{2}$ donc $\boxed{X_1 \hookrightarrow \mathscr{B}(1/2).}$

▸ Pour X_2 : on a $X_2(\Omega) = \{0,1,2\}$. De plus,

$$\begin{aligned}\mathbf{P}(X_2 = 0) &= \mathbf{P}\left({}^{C}A_1 \cap {}^{C}A_2\right)\\ &= \mathbf{P}({}^{C}A_2 | {}^{C}A_1)\,\mathbf{P}({}^{C}A_1)\\ &= \mathbf{P}({}^{C}A_2 | \alpha_1 = 1, \beta_1 = r\cdot 1)\cdot\frac{1}{2}\\ &= \frac{r}{1+r}\frac{1}{2}\\ &= \frac{r}{2(1+r)}\\ \mathbf{P}(X_2 = 2) &= \mathbf{P}(A_1 \cap A_2)\\ &= \mathbf{P}(A_2 | A_1)\,\mathbf{P}(A_1)\\ &= \mathbf{P}(A_2 | \alpha_1 = r\cdot 1, \beta_1 = 1)\cdot\frac{1}{2}\\ &= \frac{r}{1+r}\frac{1}{2}\\ &= \frac{r}{2(1+r)}.\end{aligned}$$

Donc $\mathbf{P}(X_1 = 1) = 1 - 2\frac{r}{2(1+r)} = \frac{r+1-r}{1+r} = \frac{1}{1+r}$. En conclusion :

$$\boxed{X_2(\Omega) = [\![0\,,2]\!], \quad \mathbf{P}(X_2 = 0) = \mathbf{P}(X_2 = 2) = \frac{r}{2(1+r)}, \mathbf{P}(X_2 = 1) = \frac{1}{1+r}.}$$

2. >_

```python
def simulX(n, r):
    X = 0
    a = 1
    b = 1
    for _ in range(n):
        if rd.random() < a/(a+b):
            # choix du trajet A
            X += 1
            a *= r
        else:
            # choix du trajet B
            b *= r
    return X
```

```
>>> simulX(100, 5)
1
>>> simulX(100, 1)
48
```

Pour la première exécution : le trajet choisit en premier est rapidement privilégié ; on observe donc des valeurs extrêmes pour X (proches de 100 ou proches de 0 en fonction du premier trajet choisi). Pour la seconde : l'équiprobabilité est maintenue, donc on observe des valeurs proches de $\frac{n}{2}$.

3. L'évènement $\{X_n = n\}$ signifie que toutes les fourmis ont choisi le même trajet A. Donc

$$\mathbf{P}(X_n = n) = \mathbf{P}(A_1 \cap \ldots A_n).$$

Donc par formule des probabilités composées :

$$\begin{aligned}\mathbf{P}(X_n = n) &= \mathbf{P}(A_n | A_1 \cap \ldots A_{n-1}) \cdot \ldots \mathbf{P}(A_{n-1} | A_1 \cap \ldots A_{n-2}) \cdot \ldots \mathbf{P}(A_2 | A_1)\, \mathbf{P}(A_1) \\ &= \frac{r^{n-1}}{1+r^{n-1}} \cdot \cdots \cdot \frac{r}{1+r} \cdot \frac{1}{2} \\ &= \boxed{\prod_{i=0}^{n-1} \frac{r^i}{1+r^i}.}\end{aligned}$$

Avec les notations introduites dans la suite, on a :

$$\boxed{\mathbf{P}(X_n = n) = p_{n-1}(r) \cdot \frac{1}{2}.}$$

4. On a des produits, des calculs de quotients sont donc plus adaptés pour la monotonie. Soit $n \in \mathbf{N}$, alors comme $p_n(r) \neq 0$ on a

$$\frac{p_{n+1}(r)}{p_n(r)} = \frac{r^{n+1}}{1+r^{n+1}} < 1.$$

La suite $(p_n(r))$ est donc décroissante, et minorée par zéro donc converge vers une limite finie $p(r)$. De-même :

$$\frac{q_{n+1}(r)}{q_n(r)} = 1 - \frac{1}{r^{2n+3}} < 1,$$

donc $(q_n(r))$ est décroissante, et minorée par zéro donc converge vers une limite finie $q(r)$.

$$\boxed{p_n(r) \xrightarrow{n\to\infty} p(r), \quad q_n(r) \xrightarrow{n\to\infty} q(r).}$$

5. Soit $n \geqslant 1$. En divisant les fractions au numérateur et dénominateur par $r, r^2, \ldots r^n$, on déduit :

$$\boxed{p_n(r) = \frac{1}{1+r^{-1}} \cdots \frac{1}{1+r^{-n}}},$$

donc en utilisant la majoration admise, dont la version inversée est :

$$\forall x > -1, \quad (1+x)^{-1} \geqslant e^{-x},$$

on déduit que

$$\begin{aligned}p_n(r) &= \frac{1}{1+r^{-1}} \cdots \frac{1}{1+r^{-n}} \\ &= \prod_{k=1}^{n} (1+r^{-k})^{-1} \\ &\geqslant \prod_{k=1}^{n} e^{-r^{-k}} \\ &= e^{-\sum_{k=1}^{n} (r^{-1})^k} = e^{-r^{-1} \frac{1-(r^{-1})^n}{1-r^{-1}}} \\ &\geqslant e^{-\frac{1}{r-1}}.\end{aligned}$$

somme géométrique et propriété de l'exp

$(r^{-1})^n \geqslant 0$

Donc en faisant $n \to \infty$: $\boxed{p(r) \geqslant \exp\left(-\frac{1}{r-1}\right).}$

6. On procède de la même manière.

$$\begin{aligned}q_n(r) &= \prod_{k=0}^{n} \left(1 - \frac{1}{r^{2k+1}}\right) \\ &\leqslant \prod_{k=0}^{n} e^{-\frac{1}{r^{2k+1}}} \\ &= e^{-\frac{1}{r} \sum_{k=0}^{n} \left(\frac{1}{r^2}\right)^k} = e^{-\frac{1}{r} \frac{1-\left(\frac{1}{r^2}\right)^{n+1}}{1-\frac{1}{r^2}}} \\ &\geqslant e^{-\frac{1}{r} \frac{1}{1-\frac{1}{r^2}}} = e^{-\frac{r}{r^2-1}}.\end{aligned}$$

majoration admise

somme géométrique et propriété de l'exp

$\left(\frac{1}{r^2}\right)^{n+1} \geqslant 0$

Donc en faisant $n \to \infty$: $\boxed{q(r) \leqslant \exp\left(-\frac{r}{r^2-1}\right).}$

7. On admet que $\forall r > 1$, $p(r) = q(r)$. On a déjà constaté que :

$$\forall n \geqslant 1, \quad \mathbf{P}(X_n = n) = p_{n-1}(r) \cdot \frac{1}{2}.$$

Donc la suite $(\mathbf{P}(X_n = n))$ converge vers $\frac{p(r)}{2}$. D'après les questions précédentes :

$$\boxed{\frac{1}{2} \exp\left(-\frac{1}{r-1}\right) \leqslant \lim_{n\to\infty} \mathbf{P}(X_n = n) \leqslant \frac{1}{2} \exp\left(-\frac{r}{r^2-1}\right).}$$

8. (Conclusion :) Lorsque r se rapproche de 1, on est censés retomber sur un contexte d'équiprobabilité, puisque les trajets déjà empruntés sont très peu renforcés en phéromone, donc la probabilité que sur n passages les fourmis choisissent toutes le même trajet devrait tendre vers zéro. Or, $\lim_{r\to1^+}\left(\frac{1}{2}\exp\left(-\frac{1}{r-1}\right)\right)=0$, $\lim_{r\to1^+}\left(\frac{1}{2}\exp\left(-\frac{r}{r^2-1}\right)\right)=0$, ce qui est donc cohérent avec l'intuition. Lorsque r devient grand, un trajet choisi est très rapidement renforcé en phéromone, et toutes les autres devraient suivre ce même trajet ensuite, avec forte probabilité : tout dépend donc du trajet emprunté par la toute première fourmi, A est choisi avec probabilité $\frac{1}{2}$. Mais,

$$\lim_{r\to\infty}\left(\frac{1}{2}\exp\left(-\frac{1}{r-1}\right)\right)=\frac{1}{2},\quad \lim_{r\to\infty}\left(\frac{1}{2}\exp\left(-\frac{r}{r^2-1}\right)\right)=\frac{1}{2}.$$

Là encore, c'est cohérent.

Solution (problème 1)

J'adopte la même convention que le sujet original, à savoir que les éléments de $\boldsymbol{R}^2$ sont notés en colonne.

Partie I — Modélisation et commande du processus Rappelons le système différentiel satisfait par (T_1, T_2) :

$$\begin{cases} m_1C_1\dfrac{dT_1}{dt} = h_1S_1(T_2-T_1) \\ m_2C_2\dfrac{dT_2}{dt} = h_1S_1(T_1-T_2)+h_2S_2(T_\infty-T_2)+u \end{cases} \tag{E}$$

I — 1) Commande en boucle ouvert (cas u constant)

1. On note $x_1 = T_1 - T_\infty$ et $x_2 = T_2 - T_\infty$. Alors

$$\begin{aligned} m_1C_1\frac{dx_1}{dt} &= m_1C_1\frac{dT_1}{dt} \\ &\underset{\text{en utilisant (E)}}{=} h_1S_1(T_2-T_1) \\ &= h_1S_1(T_2-T_\infty+T_\infty-T_1) \\ &= h_1S_1(x_2-x_1). \end{aligned}$$

De-même, nous obtenons pour x_2 :

$$\begin{aligned} m_2C_2\frac{dx_2}{dt} &= m_2C_2\frac{dT_2}{dt} \\ &\underset{\text{en utilisant (E)}}{=} h_1S_1(T_1-T_2)-h_2S_2x_2+u \\ &= h_1S_1(x_1-x_2)-h_2S_2x_2+u. \end{aligned}$$

En résumé, nous avons obtenu :

$$\boxed{\begin{aligned} \frac{dx_1}{dt} &= \frac{h_1S_1}{m_1C_1}(x_2-x_1) \underset{\textbf{(défi.)}}{=} f_1(x_1,x_2,u), \\ \frac{dx_2}{dt} &= \frac{h_1S_1}{m_2C_2}(x_1-x_2)-\frac{h_2S_2}{m_2C_2}x_2+\frac{1}{m_2C_2}u \underset{\textbf{(défi.)}}{=} f_2(x_1,x_2,u). \end{aligned}}$$

On obtient alors que $x=\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ vérifie le système différentiel suivant (la dérivée étant comprise ligne par ligne) :

$$\frac{dx}{dt} = Ax + bu, \tag{0.1}$$

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 $\mathfrak{M}_{2,2}(\mathbf{R})$, b un vecteur de $\mathbf{R}^2$ et $u\in\mathbf{R}$ constant.

2. **2.1)** On résout simplement le système associé. En effet, x vérifie $Ax + bu = 0$ si, et seulement si :

$$\begin{cases} -5x_1 + 5x_2 + 0 & = 0, \\ 4x_1 - 6x_2 + 9u & = 0 \end{cases} \iff \begin{cases} x_1 & = x_2, \\ 2x_1 & = 9u \end{cases}.$$

En particulier, $\boxed{x_1 = x_2.}$

2.2) On peut proposer une interprétation algébrique, en l'occurence, $\boxed{\text{l'ensemble des solutions } x \in \mathbf{R}^2 \text{ du système avec second membre } Ax = -bu \text{ est inclus dans } \mathrm{Vect}\begin{pmatrix} 1 \\ 1 \end{pmatrix}.}$

Par rapport aux solutions $t \in \mathbf{R}^+ \longmapsto x(t) = (x_1(t), x_2(t))$, on peut dire que s'il existe $t_0 \in \mathbf{R}^+$ tel que $\frac{dx}{dt}(t_0) = 0$, alors $x_1(t_0) = x_2(t_0)$. Donc $\boxed{\text{les temps d'annulation de la dérivée de } x \text{ ne peuvent être que des temps où la température de l'eau est égale à celle des petits poids.}}$[3]

2.3) Calculons la valeur de u si on considère que $x_1 = x_2 = 72$. D'après la seconde ligne du système, si $x_1 = x_2 = 72$, alors $u = \frac{2}{9}x_1 = \frac{2}{9}72 \boxed{= 16.}$

3. **3.1)** Soit $\lambda \in \mathbf{C}$. Alors $\det(A - \lambda I_2) = \begin{vmatrix} -5-\lambda & 5 \\ 4 & -6-\lambda \end{vmatrix} = \lambda^2 + 11\lambda + 10$. Le discriminant associé est alors $\Delta = 81 = 9^2$, donc les deux racines sont

$$\frac{-11+9}{2} = -1 \quad \text{ou} \quad \frac{-11-9}{2} = -10.$$

Donc $\boxed{\mathrm{Spec}(A) = \{-10, -1\}.}$

3.2) La matrice A est de format 2×2, et possède deux valeurs propres distinctes, donc $\boxed{\text{est diagonalisable (sur } \mathbf{R}).}$

3.3) Commençons par calculer les éléments de $E_{-10}(A)$. Soit $X = \begin{pmatrix} x \\ y \end{pmatrix} \in \mathfrak{M}_{2,1}(\mathbf{R})$,

$$\begin{aligned} (A + 10I_2)X = 0 & \iff \begin{cases} 5x + 5y & = 0 \\ 4x + 4y & = 0 \end{cases} \\ & \iff x = -y. \end{aligned}$$

Donc $\boxed{E_{-10}(A) = \mathrm{Vect}\begin{pmatrix} 1 \\ -1 \end{pmatrix}.}$ Faisons de-même pour l'autre. Soit $X = \begin{pmatrix} x \\ y \end{pmatrix} \in \mathfrak{M}_{2,1}(\mathbf{R})$,

$$\begin{aligned} (A + I_2)X = 0 & \iff \begin{cases} -4x + 5y & = 0 \\ 4x - 5y & = 0 \end{cases} \\ & \iff 4x = 5y. \end{aligned}$$

Donc $\boxed{E_{-1}(A) = \mathrm{Vect}\begin{pmatrix} 1 \\ \frac{4}{5} \end{pmatrix}.}$

3.4) Notons $\boxed{P = \begin{pmatrix} 1 & 1 \\ -1 & \frac{4}{5} \end{pmatrix}}$, alors P est inversible d'inverse $\boxed{P^{-1} = \frac{5}{9}\begin{pmatrix} \frac{4}{5} & -1 \\ 1 & 1 \end{pmatrix}.}$ D'après la formule de changement de base, et puisque A est diagonalisable, nous avons

$$\boxed{A = PDP^{-1}, \quad \text{avec} \quad D = \mathrm{Diag}(-10, -1).}$$

[3] avec CAUCHY-Lipschitz, on pourrait dire beaucoup plus....

4. Soit x une solution de (1). On définit $z = \mathrm{P}^{-1}x$, où $z = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix}$.

4.1) Notons $z = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix}$. Alors $\mathrm{P}^{-1}x = \frac{5}{9}\begin{pmatrix} \frac{4}{5} & -1 \\ 1 & 1 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \frac{5}{9}\begin{pmatrix} \frac{4}{5}x_1 - x_2 \\ x_1 + x_2 \end{pmatrix}$.

Donc, d'après les règles d'opérations sur les dérivées des fonctions à valeurs réelles, nous avons :

$$\frac{\mathrm{d}z}{\mathrm{d}t} = \frac{5}{9}\begin{pmatrix} \frac{4}{5}x_1' - x_2' \\ x_1' + x_2' \end{pmatrix} = \mathrm{P}^{-1}\begin{pmatrix} x_1' \\ x_2' \end{pmatrix} = \boxed{\mathrm{P}^{-1}\frac{\mathrm{d}x}{\mathrm{d}t}}.$$

4.2) Nous obtenons alors :

$$\begin{aligned} \frac{\mathrm{d}z}{\mathrm{d}t} &= \mathrm{P}^{-1}\frac{\mathrm{d}x}{\mathrm{d}t} \\ &= \mathrm{P}^{-1}(\mathrm{PDP}^{-1}x + bu) && \textit{ED vérifiée par } x \\ &= \mathrm{DP}^{-1}x + \mathrm{P}^{-1}bu \\ &= \boxed{\mathrm{D}z + \beta u, \quad \text{en posant } \beta = \mathrm{P}^{-1}b.} && \textit{définition de } z \end{aligned}$$

4.3) On suppose que $\mathrm{T}_1(0) = 20^\circ\mathrm{C}$ et $\mathrm{T}_2(0) = 92^\circ\mathrm{C}$. Alors :

$$z(0) = \mathrm{P}^{-1}x(0) = \frac{5}{9}\begin{pmatrix} \frac{4}{5} & -1 \\ 1 & 1 \end{pmatrix}\begin{pmatrix} 0 \\ 72 \end{pmatrix} = \frac{5}{9}\begin{pmatrix} -72 \\ 72 \end{pmatrix} = \boxed{40\begin{pmatrix} -1 \\ 1 \end{pmatrix}.}$$

4.4) Calculons pour commencer le vecteur β.

$$\beta = \frac{5}{9}\begin{pmatrix} \frac{4}{5} & -1 \\ 1 & 1 \end{pmatrix}\begin{pmatrix} 0 \\ 9 \end{pmatrix} = 5\begin{pmatrix} \frac{4}{5} & -1 \\ 1 & 1 \end{pmatrix}\begin{pmatrix} 0 \\ 1 \end{pmatrix} = 5\begin{pmatrix} -1 \\ 1 \end{pmatrix}.$$

L'équation (2) est équivalente à la suivante[4] :

$$\begin{cases} z_1'(t) &= -10z_1(t) - 5.16, \\ z_2'(t) &= -z_2(t) + 5.16 \end{cases}$$

$$\begin{cases} z_1(t) &= \mathrm{K}e^{-10t} - 8, \\ z_2(t) &= \mathrm{K}'e^{-t} + 5\times 16 \end{cases}, \quad \mathrm{K}, \mathrm{K}' \in \mathbf{R}.$$

En tenant compte des conditions initiales $z_1(0) = -40$, $z_2(0) = 40$, on obtient comme conditions

$$\mathrm{K} - 8 = -40, \quad \mathrm{K}' + 5.16 = 40,$$

d'où finalement, pour tout $t \in \mathbf{R}^+$:

$$\boxed{z_1(t) = -32e^{-10t} - 8, \quad z_2(t) = -40e^{-t} + 5.16.}$$

4.5) En utilisant la relation $x = \mathrm{P}z$, on obtient pour tout $t \in \mathbf{R}^+$:

$$\begin{cases} x_1(t) &= z_1(t) + z_2(t) \\ x_2(t) &= -z_1(t) + \frac{4}{5}z_2(t). \end{cases}$$

Après calculs, on trouve le système de l'énoncé :

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

5. 5.1) Soit $t \in \mathbf{R}^+$. Alors la fonction x_2 est dérivable sur $\mathbf{R}^+$ en tant que somme de telles fonctions. Et,

$$x_2'(t) = 32e^{-10t} - 320e^{-10t}.$$

[4] version « décorrélée » du système initial

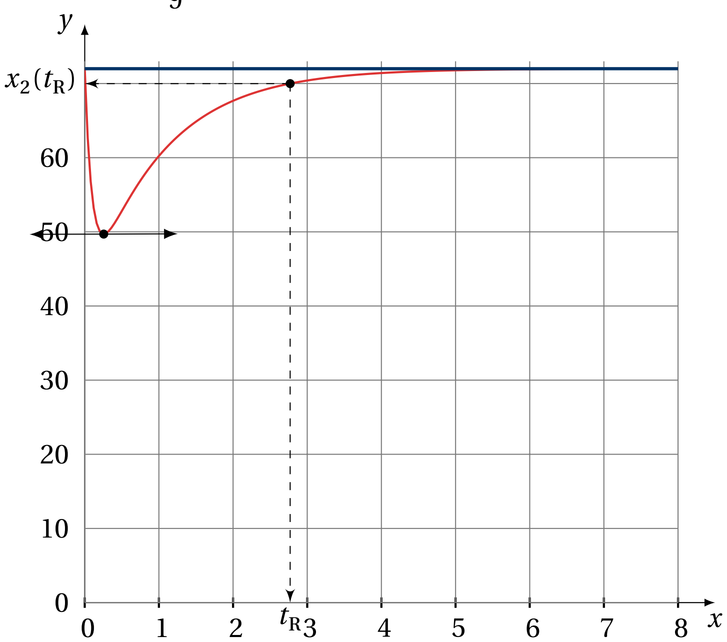
Dès lors, nous avons

$$\begin{aligned} x_2'(t) \geqslant 0 &\iff 32\mathrm{e}^{-t} \geqslant 320\mathrm{e}^{-10t} \\ &\iff -t \geqslant \ln 10 - 10t \\ &\iff 9t \geqslant \ln 10 \\ &\iff t \geqslant \frac{\ln 10}{9} (\geqslant 0). \end{aligned}$$

On déduit alors le tableau de variations ci-dessous, puis la courbe représentative de la fonction.

t	0		$\frac{\ln 10}{9}$		$+\infty$
$x_2(t)$	72	$\searrow$	$x_2\left(\frac{\ln 10}{9}\right)$	$\nearrow$	72

La dérivée s'annulant en $\frac{\ln 10}{9}$, nous avons donc une pente horizontale en ce point.



5.2) Le processus commence avec une eau très chaude (92°C), on démarre ensuite la chauffe qui au début ne parvient pas à compenser la chute de température due à la perdition d'énergie. Dans un troisième temps, la température de l'eau augmente, conséquence du démarrage de la chauffe.

5.3)

$$t_R = \min\{t \geqslant 0\,;\ \forall t' \geqslant t,\ T_2(t') \geqslant 90\} = \min\{t \geqslant 0\,;\ \forall t' \geqslant t,\ x_2(t') \geqslant 70\}.$$

C'est donc le plus petit temps à partir duquel $\boxed{\text{la courbe de } x_2 \text{ reste après au-dessus de la droite } x = 70.}$ Ainsi, après ce temps, la courbe de chauffe x_2 respecte le cahier des charges imposé par l'industriel.

5.4) Nous avons $x_2(\ln(16)) = 72 - \frac{32}{16} + \frac{32}{16^{10}} = 70 + \frac{32}{16^{10}} \approx \boxed{70}$.

Donc $\boxed{\ln(16) \text{ est une bonne approximation de } t_R.}$ Les valeurs propres sont les coefficients dans les exponentielles décroissantes. $\boxed{\text{Plus les valeurs propres seront négatives, plus } x_2 \text{ convergera vite vers 72, donc plus } t_R \text{ sera petit.}}$

I — 2) Commande en boucle fermée (cas u variable)

6. Avec ce choix de chauffage, lorsque x_2 baisse alors u augmente. Ainsi, $\boxed{\text{ce système aura tendance à réguler la température de l'eau}}$ en l'augmentant/diminuant lorsque cela est nécessaire.

7. Si $k = 0$, alors u est constant et on tombe dans le cadre des questions précédentes. Ainsi : $\boxed{t_R \approx \ln(16)}$.

8. On définit $w(t) = x(t) - \begin{pmatrix} 72 \\ 72 \end{pmatrix}$ pour tout $t \geqslant 0$. Constatons que ${}^\top c.w =$

$\begin{pmatrix} 0 & 1 \end{pmatrix}\begin{pmatrix} x_1-72 \\ x_2-72 \end{pmatrix} = x_2 - 72$. On en déduit alors la dérivée de w,

$$\begin{aligned}
\frac{\mathrm{d}w}{\mathrm{d}t} &= \frac{\mathrm{d}x}{\mathrm{d}t} \\
&= \mathrm{A}x + bu \\
&= \mathrm{A}\left(w + 72\begin{pmatrix} 1 \\ 1 \end{pmatrix}\right) + b(16 + k(72 - x_2)) \\
&= \mathrm{A}\left(w + 72\begin{pmatrix} 1 \\ 1 \end{pmatrix}\right) + b(16 - k^\top c.w) \\
&= (\mathrm{A} - kb^\top c)w + 72\mathrm{A}\begin{pmatrix} 1 \\ 1 \end{pmatrix} + 16b
\end{aligned}$$

Après calculs, on trouve que $72\mathrm{A}\begin{pmatrix} 1 \\ 1 \end{pmatrix} + 16b = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$, d'où la formule de l'énoncé :

$$\boxed{\frac{\mathrm{d}w}{\mathrm{d}t} = (\mathrm{A} - kb^\top c)w}.$$

9. On note $\lambda_1(k)$ et $\lambda_2(k)$ les valeurs propres de $\mathrm{A} - kb^\top c$ telles que $\lambda_1(k) < \lambda_2(k)$.

9.1) Calculons les deux valeurs propres à l'aide du déterminant. Nous avons

$$\mathrm{A} - kb^\top c = \begin{pmatrix} -5 & 5 \\ 4 & -6 \end{pmatrix} - k\begin{pmatrix} 0 \\ 9 \end{pmatrix}\begin{pmatrix} 0 & 1 \end{pmatrix} = \begin{pmatrix} -5 & 5 \\ 4 & -6 \end{pmatrix} - k\begin{pmatrix} 0 & 0 \\ 0 & 9 \end{pmatrix}.$$

Soit $\lambda \in \mathbf{C}$, alors calculons

$$\mathrm{D}(\lambda) \underset{\textbf{(défi.)}}{=} \begin{vmatrix} -5-\lambda & 5 \\ 4 & -6-9k-\lambda \end{vmatrix} = (-5-\lambda)(-6-9k-\lambda) - 20.$$

En développant, on obtient :

$$\mathrm{D}(\lambda) = \lambda^2 + \lambda(9k+11) + 5(9k+2).$$

Le discriminant de ce polynôme en λ est alors $(9k+11)^2 - 20(9k+2) = 9(9k^2+2k+9) \geqslant 0$ puisque $k \geqslant 0$. Ainsi,

$$\boxed{\begin{aligned}&\mathrm{Spec}(\mathrm{A} - kb^\top c) \\ &= \left\{ \frac{-9k-11-3\sqrt{9k^2+2k+9}}{2}, \frac{-9k-11+3\sqrt{9k^2+2k+9}}{2} \right\}.\end{aligned}}$$

On note alors $\lambda_1(k) = \dfrac{-9k-11-3\sqrt{9k^2+2k+9}}{2}$ et $\lambda_2(k) = \dfrac{-9k-11+3\sqrt{9k^2+2k+9}}{2}$.

9.2) La fonction λ_1 est une somme de fonctions décroissantes, donc $\boxed{\lambda_1 \text{ est décroissante.}}$ Pour λ_2, nous voilà condamnés à faire les calculs à la main en calculant la dérivée. Soit $k \in \mathbf{R}^+$,

$$\begin{aligned}
\lambda_2'(k) &= \frac{1}{2}\left(-9 + \frac{3}{2}\frac{18k+2}{\sqrt{9k^2+2k+9}}\right) \\
&= \frac{1}{2}\left(-9 + 3\frac{9k+1}{\sqrt{9k^2+2k+9}}\right) \\
&= \frac{1}{2}\frac{-9\sqrt{9k^2+2k+9} + 3(9k+1)}{2\sqrt{9k^2+2k+9}}
\end{aligned}$$

Ainsi, on a

$$\begin{aligned}
&\lambda_2'(k) \leqslant 0 \\
\iff\quad & 3(9k+1) \leqslant 9\sqrt{9k^2+2k+9} \\
\iff\quad & (9k+1)^2 \leqslant 3^2(9k^2+2k+9) \quad \textit{quantités positives car } k \geqslant 0 \\
\iff\quad & 81k^2 + 18k + 1 \leqslant 81k^2 + 18k + 81 \\
\iff\quad & 1 \leqslant 81.
\end{aligned}$$

Ainsi la fonction λ_2' est négative sur $\mathbf{R}^+$, d'où $\boxed{\text{la décroissance de } \lambda_2 \text{ sur } \mathbf{R}^+.}$

10. Plus k, plus les valeurs propres de $A - k b^\top c$ seront petites dans les négatifs, donc plus le temps t_R sera petit d'après la question 5.4. En revanche, si l'on augmente trop la valeur de k, on augmente l'énergie à fournir pour la chauffe.

Partie II — 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 comporte des rappels sur les commandes utiles. Avant chaque algorithme, on écrira brièvement le raisonnement suivi et la formule qu'il est censé calculer.

II — 1) Recherche du minimum d'une fonction

Soit $f : t \in \mathbf{R} \longrightarrow 72 - 32e^{-t} + 32e^{-10t}$.

1.

```python
import numpy as np
def f(t):
    return 72 - 32*np.exp(-t) + 32*np.exp(-10*t)
```

Une déclaration alternative serait simplement

```python
f = lambda t:72 - np.exp(-t) + 32*np.exp(-10*t)
```

2. Après les lignes 2 et 5, la variable `Lt1` contient les éléments d'une subdivision régulière en N points de l'intervalle $[0,5]$, *i.e.* les $\boxed{\dfrac{5k}{N} \text{ avec } k \in [\![0\,, N]\!].}$ La variable `Ly1` contient les images par f de cette subdivision sauf le dernier, donc les $\boxed{f\left(\dfrac{5k}{N}\right) \text{ avec } k \in [\![0\,, N-1]\!].}$

3. Il suffit alors d'effectuer une recherche de minimum sur les éléments de la liste `Ly1`.

```python
def minimum_f(N):
    Lt1 = np.linspace(0, 5, N)
    Ly1 = []
    for k in range(0, N):
        Ly1.append(f(Lt1[k]))
    # Recherche de minimum
    mini = Ly1[0]
    ind_mini = 0
    for i in range(1, N):
        if Ly1[i] < mini:
            mini = Ly1[i]
            ind_mini = i
    return [mini, Lt1[ind_mini]]
```

Plus précisément, la deuxième coordonnée du résultat sera la dernière valeur où le minimum est atteint. Testons par exemple avec un jeu de valeurs donné par l'énoncé : pour $N_1 = 20$, `minimum_f(20)` retourne bien

$$[49.70701495513559, 0.26315789473684 21].$$

4. **4.1)** N_2 est bien supérieur à N_1. L'algorithme de calcul d'une valeur approchée du minimum risque d'être plus précis en utilisant N_2. La différence provient donc d'une différence dans les précisions choisies (dans le calcul de la subdivision de $[0,5]$).

4.2) Logiquement (mais encore faudrait-il le montrer), la valeur t_{m_2} associée à N_2 devrait être meilleure que t_{m_1}.

5. L'algorithme cherche à approcher $t^* = \dfrac{\ln 10}{9} \approx 0.2558$. Nous avons obtenu $m_2 \leqslant m_1 \leqslant m_3$ et $t_{m_2} \leqslant t^* \leqslant t_{m_1} \leqslant t_{m_3}$. Il n'est donc pas clair que la suite des erreurs commises $\left(\left|t_{m_N} - t^*\right|\right)_N$ est décroissante en N (la valeur obtenue pour $N = 50$ semble moins précise que celle pour $N = 20$), néanmoins cette suite semble converger vers zéro lorsque $N \longrightarrow \infty$ d'après le résultat de l'appel associé à $N = 20000$.

II — 2) Recherche du point d'annulation d'une fonction

Remarque : dans le cas où il existe deux solutions, la suite (u_n) n'est pas bien définie pour certains u_0

6. On construit les termes de la suite récurrente jusqu'à ce que la condition de terminaison $\left|F(u_n)\right| < e$ soit satisfaite.

```python
def Newt(F, G, u0, e):
    """
    retourne le terme de la suite de Newton à précision e
    """
    u = u0
    while np.abs(F(u)) >= e:
        u = u - F(u)/G(u)
    return u
```

Par exemple, si l'on considère $F(x) = x^2 - 2$ et $G(x) = 2x$ pour $x \in I = \mathbf{R}^+$, alors l'algorithme devrait retourner une valeur approchée de $\sqrt{2}$.

```python
def F(x):
    return x**2 - 2
def G(x):
    return 2*x
```

L'appel `Newt(F, G, 1, 10**(-4))` retourne 1.4142156862745099. C'est une bonne approximation de $\sqrt{2}$.

7. Tester l'égalité à zéro d'un flottant n'est absolument pas pertinent. On remplace donc cette condition d'arrêt par une majoration par une quantité très petite. Plus celle-ci est petite, meilleure sera l'approximation de x vérifiant $F(x) = 0$. $\boxed{\text{On choisira donc plutôt } e_3 = 10^{-8}}$.

8. Cela n'est pas pertinent, toujours du fait de la gestion des flottants par Python puisque des valeurs approchées sont utilisées à chaque étape de l'algorithme.

9. Si un tel n_0 n'existe pas, alors la structure `while` ne s'arrêtera pas. On pourrait donc ajouter une borne sur le nombre d'itérations, c'est ce qui est fait dans la suite.

10. On souhaite modifier la fonction `Newt` de manière à fixer un nombre maximal d'itérations pour la boucle `while`. Il suffit alors d'ajouter une seconde condition d'arrêt.

```python
def NewtS(F, G, u0, e, nder):
    """
    retourne le terme de la suite de Newton à précision e avec
    ↪ borne sur le nombre d'itérations
    """
    u = u0
    n = 0 # on ajoute un compteur d'itérations
    while np.abs(F(u)) >= e and n <= nder:
        n += 1
        u = u - F(u)/G(u)
    if n <= nder:
        return u
    else:
        return False
```

Reprenons l'exemple de $\sqrt{2}$ mentionné ci-dessus. L'appel `NewtS(F, G, 1, 10**(-4), 10)` retourne 1.4142156862745099.

11. D'après le tableau de variations établi pour x_2, le temps t_R peut être définie comme le temps $t \geqslant t^* = \dfrac{\ln 10}{9}$ pour lequel $x_2(t) = 70$. Notons $F \left| \begin{array}{ccc} [t^*, \infty] & \longrightarrow & \mathbf{R} \\ t & \longmapsto & x_2(t) - 70 \end{array} \right.$.

$\boxed{\text{La méthode de Newton appliquée à la fonction F permettrait d'obtenir une valeur approchée de } t_R.}$

Il faudrait cependant veiller à l'initialisation pour garantir la convergence vers t_R et pas vers l'autre solution.

12. Constatons tout d'abord que pour tout $t \in \mathbf{R}^+$,

$$\frac{dF_1}{dt}(t) = f'(t) = G_1(t),$$

où F_1, G_1 sont les deux fonctions mathématiques associées aux fonctions Python F_1 et G_1.

$\boxed{\text{Les appels retournent donc, sous réserve de convergence, une valeur approchée d'une solution de l'équation } x_2(t) = 70}$.

Pour le premier appel, vu le résultat, on constate que c'est une valeur approchée de t_R qui est retournée. En revanche, pour le second, c'est la solution sur l'intervalle

$\left[0, \dfrac{\ln 10}{9}\right]$. La différence observée est due à l'initialisation.