ÉPREUVE SPÉCIFIQUE - FILIÈRE TPC

MODÉLISATION

Durée : 4 heures

N.B. : le candidat attachera la plus grande importance à la clarté, à la précision et à la concision de la rédaction. Si un candidat est amené à repérer ce qui peut lui sembler être une erreur d'énoncé, il le signalera sur sa copie et devra poursuivre sa composition en expliquant les raisons des initiatives qu'il a été amené à prendre.

RAPPEL DES CONSIGNES

- *Utiliser uniquement un stylo noir ou bleu foncé non effaçable pour la rédaction de votre composition ; d'autres couleurs, excepté le vert, bleu clair ou turquoise, peuvent être utilisées, mais exclusivement pour les schémas et la mise en évidence des résultats.*
- *Ne pas utiliser de correcteur.*
- *Écrire le mot FIN à la fin de votre composition.*

Les calculatrices sont autorisées.

Le sujet est composé de trois parties indépendantes et d'une Annexe en fin de sujet.

En voiture !

Partie I - Principe des capteurs pneumatiques

Pour détecter le passage de véhicules sur la route, on place un tube en caoutchouc rempli d'air perpendiculairement à la chaussée. Le tube est bouché à une extrémité et relié à un capteur à l'autre extrémité. Lorsqu'un véhicule roule sur le tube, une onde de pression se propage dans le tube jusqu'au détecteur qui comptabilise le passage (**figure 1**).

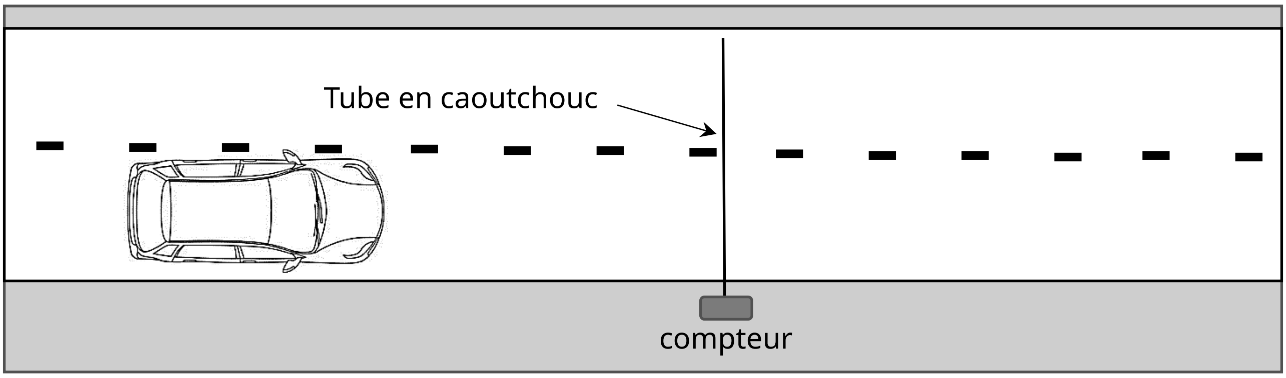

Figure 1 - Capteur pneumatique

Q1. Expliquer pourquoi il est recommandé de changer le tube tous les 15 jours.

I.1 - Propagation des ondes acoustiques dans un tuyau souple

On considère un tube en caoutchouc de section circulaire A_0 et d'axe Ox rempli d'air (**figure 2**). Au repos, l'air a une masse volumique μ_0, une pression P_0 et le champ des vitesses est nul.

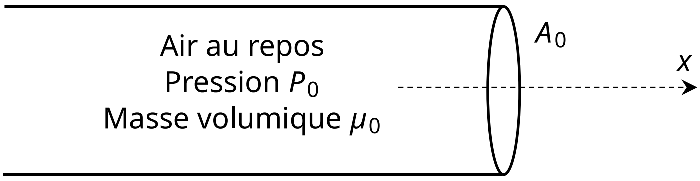

Figure 2 - Tuyau de section A_0 rempli d'air au repos

On s'intéresse à la propagation de perturbations le long de l'axe Ox. Dans le cadre de l'*approximation acoustique*, les champs de vitesse, de pression et de masse volumique s'expriment sous la forme :

$$\begin{cases} v(x,t) & \text{appelée la } \textit{vitesse acoustique}, \\ P(x,t) = P_0 + p_1(x,t) & \text{où } |p_1(x,t)| \ll P_0 \text{ est la } \textit{surpression acoustique}, \\ \mu(x,t) = \mu_0 + \mu_1(x,t) & \text{où } |\mu_1(x,t)| \ll \mu_0. \end{cases}$$

Ces grandeurs sont uniformes sur une section du tube et les effets de la pesanteur sont négligés. L'air est assimilé à un gaz parfait en évolution isentropique.

Compressibilité isentropique de l'air

On note χ_S le coefficient de *compressibilité isentropique* de l'air : $\chi_S = \dfrac{1}{\mu(x,t)} \left(\dfrac{\partial \mu(x,t)}{\partial P(x,t)} \right)_S$.

Q2. En linéarisant le coefficient de compressibilité isentropique χ_S, montrer que :

$$\mu_1 = \mu_0 \chi_S \, p_1. \qquad \textbf{(1)}$$

Distensibilité isentropique du tube

Sous l'effet de la variation de la pression dans le tuyau, celui-ci se déforme. Sa section $A(x,t)$ varie et on la note : $A(x,t) = A_0 + a_1(x,t)$, où $|a_1(x,t)| \ll A_0$.

On utilise le coefficient de *distensibilité isentropique* $D_S = \dfrac{1}{A(x,t)}\left(\dfrac{\partial A(x,t)}{\partial P(x,t)}\right)_S$ pour quantifier la variation de la section du tuyau lors du passage de l'onde acoustique.

Q3. En linéarisant le coefficient de distensibilité isentropique D_S, montrer que :

$$a_1 = A_0\, D_S\, p_1. \qquad \textbf{(2)}$$

Équation de conservation de la masse

On étudie, entre les dates t et $t + \mathrm{d}t$, la tranche d'air située entre les abscisses x et $x + \mathrm{d}x$ (**figure 3**).

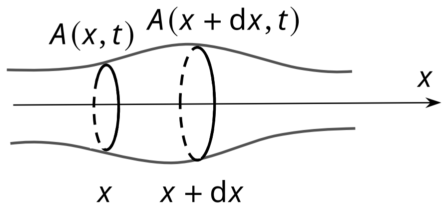

Figure 3 - Système étudié : tranche d'air d'épaisseur dx

Q4. Exprimer la masse $\mathrm{d}m(t)$ de ce système à la date t en fonction de $A(x,t)$, $\mu(x,t)$ et de $\mathrm{d}x$.
De la même manière, exprimer la masse $\mathrm{d}m(t+\mathrm{d}t)$ à la date $t + \mathrm{d}t$.

Q5. Exprimer la masse δm_e de fluide entrant dans ce système à l'abscisse x pendant la durée dt en fonction de $\mu(x,t)$, $v(x,t)$, $A(x,t)$ et de dt.
De la même façon, exprimer la masse δm_s sortant de ce système à l'abscisse $x + \mathrm{d}x$ pendant la durée dt.

Q6. En réalisant un bilan de masse sur le système considéré, établir avec soin l'équation de conservation de la masse :

$$\frac{\partial}{\partial x}\big[\mu(x,t)A(x,t)v(x,t)\big] + \frac{\partial}{\partial t}\big[\mu(x,t)A(x,t)\big] = 0.$$

Q7. En se limitant aux termes d'ordre 1, montrer que l'on obtient l'équation linéarisée :

$$\mu_0 A_0 \frac{\partial v(x,t)}{\partial x} + \mu_0 \frac{\partial a_1(x,t)}{\partial t} + A_0 \frac{\partial \mu_1(x,t)}{\partial t} = 0. \qquad \textbf{(3)}$$

Équation de propagation des ondes sonores dans le tube souple

Q8. À l'aide des relations **(1)**, **(2)** et **(3)**, établir l'équation :

$$(\chi_S + D_S)\frac{\partial p_1(x,t)}{\partial t} + \frac{\partial v(x,t)}{\partial x} = 0. \qquad \textbf{(4)}$$

On admet que l'air suit l'équation du mouvement :

$$\mu_0 \frac{\partial v(x,t)}{\partial t} = -\frac{\partial p_1(x,t)}{\partial x}. \qquad \textbf{(5)}$$

Q9. En combinant les équations **(4)** et **(5)**, montrer que la surpression $p_1(x,t)$ obéit à l'équation :

$$\frac{\partial^2 p_1}{\partial x^2} = \frac{1}{c^2}\frac{\partial^2 p_1}{\partial t^2} \qquad \textbf{(6)}$$

et déterminer l'expression de la célérité c en fonction de χ_S, D_S et de μ_0.
Donner le nom de l'**équation (6)**.

Q10. Vérifier la dimension de l'expression de c établie en fonction de χ_S, D_S et de μ_0.
Calculer c avec $\chi_S = 6{,}9 \cdot 10^{-6}\,\mathrm{Pa}^{-1}$, $D_S = 5{,}0 \cdot 10^{-6}\,\mathrm{Pa}^{-1}$ et $\mu_0 = 1{,}2\,\mathrm{kg} \cdot \mathrm{m}^{-3}$.
Comparer la célérité d'une onde dans le tube à celle d'une onde sonore dans l'air libre et commenter une éventuelle différence.

I.2 - Traitement des données numériques

Le **code Python 1** assure le traitement des données issues du comptage des véhicules et leur affichage. Les parties repérées par « `___` » seront à remplir.

```
import matplotlib.pyplot as plt
from enregistrements import mesures              # importation des données

def statistiques(mesures):
    Nbj = ___         # nombre de jours             # Voir Q12
    Nbh = ___         # nombre d'heures             # Voir Q13
    Vpj = []          # liste des nombres de véhicules par jour
    Max = (0, 0, 0)  # tuple (maximum, jour, heure)
    for j in range(___):                            # Voir Q14
        Vpj.___                                     # Voir Q15
        for h in range(Nbh):
            if ___ > Max[___]:                      # Voir Q16
                Max = (___, ___, ___)               # Voir Q16
    return Vpj, Max

def trace(mesures, jour):
    heure = [h for h in range(1, 25)]
    plt.bar(heure, ___)                             # Voir Q17
    plt.grid()
    plt.xlabel("Heure")
    plt.ylabel("Nombre de véhicules")
    plt.title("Nombre de véhicules détectés le premier jour")
    plt.show()
    return

print(statistiques(mesures))
trace(___, ___)                                     # Voir Q18
```

Code Python 1 - Traitement des données numériques

Le passage des véhicules sur la route est comptabilisé par heure et par jour. Ces données sont stockées dans le tableau `mesures`, importées par le module `enregistrements`, dont l'extrait ci-dessous montre la structure.

```
>>> from enregistrements import mesures
>>> mesures
array([[ 4,  10,  12,  17,  27,  70, 100, 219, 462, 329, 250, 150, 115,
        120, 225, 335, 425, 515, 305, 109,  96,  30,  14,   2],             # 1er jour
       [ 3,  12,  11,  20,  30,  68,  95, 218, 468, 365, 214, 156, 156,
        214, 365, 468, 218,  95,  68,  30,  20,  11,  12,   3],             # 2e jour
        ...
```

Chaque ligne du tableau `mesures` représente une journée de comptage soit 24 entiers qui correspondent chacun au nombre de véhicules recensés par heure.
Sur cet extrait, le nombre grisé 4 indique que 4 véhicules ont circulé le premier jour du recensement entre 0 et 1 heure du matin, le nombre grisé 214 indique que 214 véhicules ont circulé le deuxième jour entre 10 et 11 heures du matin, etc.

Q11. Donner la commande Python, sous la forme `mesures[___, ___]`, permettant d'afficher le nombre de véhicules détectés le 5e jour entre 13 et 14 heures.

La fonction `statistiques()` compte le nombre de véhicules passés chaque jour et détermine le jour et l'heure correspondant au maximum du nombre de véhicules ayant circulé.

Q12. Définir la variable `Nbj` à partir de la variable `mesures` permettant d'obtenir le nombre de jours que contient l'enregistrement (**ligne 5**).

Q13. Définir la variable `Nbh` à partir de la variable `mesures` permettant de déterminer le nombre d'heures par jour (**ligne 6**).

Q14. Compléter la **ligne 9** permettant de balayer les indices de tous les jours de l'enregistrement.

Q15. Compléter la **ligne 10** permettant de stocker dans la liste `Vpj` la somme du nombre de véhicules ayant circulé le jour `j`.
On utilisera 2 commandes Python figurant dans l'annexe.

On souhaite déterminer le jour et l'heure associés au passage du plus grand nombre de véhicules. Les 3 valeurs correspondantes sont stockées dans le tuple `Max`, initialisé à la **ligne 8**, dans l'ordre (maximum, jour, heure).

Q16. Compléter la **ligne 12** permettant de détecter un maximum, puis la **ligne 13** permettant d'actualiser le tuple `Max` des résultats.

La fonction `trace()` du **code Python 1** trace l'histogramme du nombre de véhicules ayant circulé lors d'une journée choisie (**figure 4**). Elle prend en paramètres le tableau `mesures` et l'entier `jour` correspondant au jour souhaité.
Une aide sur le module `matplotlib` est proposée en annexe.

Q17. En utilisant les paramètres de la fonction `trace()`, compléter la **ligne 18** permettant de tracer l'histogramme correspondant à la journée souhaitée.

Q18. Compléter l'appel de la fonction `trace()` à la **ligne 27** pour tracer l'histogramme de la 1re journée de comptage.

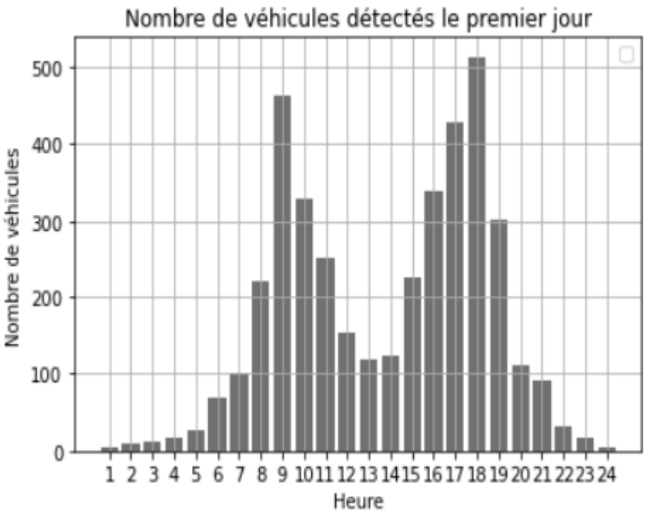

Figure 4 - Diagramme à bâtons représentant le nombre de véhicules détectés par heure le premier jour

Partie II - Modélisation microscopique d'un embouteillage routier

Les modèles microscopiques décrivent le comportement individuel de chaque véhicule en respectant les interactions entre les véhicules.
L'objectif de cette partie est d'observer l'effet d'un ralentissement sur une file de véhicules.

Figure 5 - Exemple de trafic

II.1 - Description du modèle

Considérons N véhicules identiques numérotés de $n = 0$ à $n = N - 1$ (**figure 6**). On note :

- $x_n(t)$: la *position* du véhicule n à la date t ;
- $v_n(t) = \dot{x}_n(t)$: la *vitesse* du véhicule n à la date t ;
- $d_n(t) = x_{n-1}(t) - x_n(t)$: la *distance inter-véhiculaire* séparant le véhicule n de celui qui se trouve devant lui (cette distance est positive).

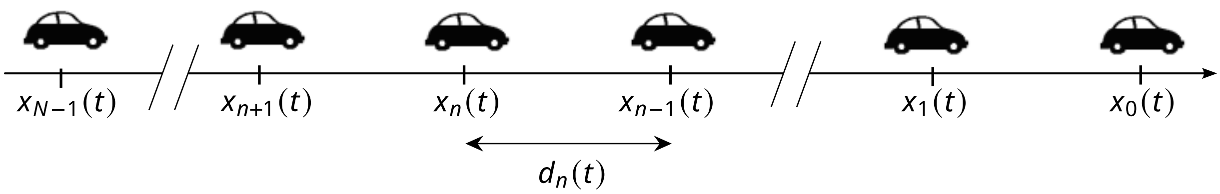

Figure 6 - Schéma du problème

On suppose que les comportements individuels sont tels que chaque conducteur n adapte sa vitesse v_n en fonction de la distance d_n qui le sépare du véhicule qui se trouve devant lui en suivant la relation :

$$v_n(t) = v_{\max} \times f(d_n(t)),$$

avec $v_{\max} = 30\,\mathrm{m\cdot s^{-1}}$ et la fonction f, telle que :

$$f(d) = \begin{cases} 0 & \text{si } d \leqslant d_{\min} \\ 1 - \exp\left(-\dfrac{d - d_{\min}}{d_0}\right) & \text{si } d \geqslant d_{\min} \end{cases} \quad \text{avec} \begin{cases} d_{\min} = 7{,}0\,\mathrm{m} \\ d_0 = 20\,\mathrm{m}. \end{cases} \qquad \textbf{(7)}$$

Q19. Donner les valeurs de $f(0)$, $f(d_{\min})$ et de $\lim\limits_{d \to \infty} f(d)$.
Tracer l'allure de f en fonction de d.

Q20. Expliquer le sens physique de la variable v_{max}.
Expliquer le sens physique de la variable d_{min}.
Comment qualifier la circulation routière lorsque $d \leqslant d_{\text{min}}$?

Trafic stationnaire

On se place en *trafic stationnaire* qui est un régime de circulation particulier dans lequel tous les véhicules se déplacent à la même vitesse v. On suppose $v = 20\,\text{m}\cdot\text{s}^{-1}$.

Q21. Montrer que la distance inter-véhiculaire d est constante.

Q22. Établir l'expression de la distance inter-véhiculaire : $d = d_{\text{min}} - d_0 \ln\left(1 - \dfrac{v}{v_{\text{max}}}\right)$.
Calculer sa valeur dans le cas du trafic stationnaire.

Pour des raisons de sécurité, le code de la route impose de laisser une distance équivalente à deux secondes de trajet entre deux véhicules qui se suivent.

Q23. Préciser si les paramètres du modèle sont compatibles avec le code de la route.

II.2 - Résolution numérique : étude de l'effet d'un ralentissement

On suppose que l'ensemble des véhicules roule à vitesse constante lorsque celui de tête ralentit puis accélère durant un court instant pour retrouver la vitesse initiale.
De proche en proche, les véhicules suivants vont devoir adapter leur vitesse, ce qui va engendrer le déplacement de la perturbation.

Pour suivre les N véhicules à M dates, on utilise les indices n et m, tels que :

- $n \in [\![0, N-1]\!]$: repère le véhicule n (position x_n, vitesse v_n) ;
- $m \in [\![0, M-1]\!]$: indique la date t_m.

À l'aide de ces indices, on pose :

- x_n^m : la position du véhicule n à la date t_m ;
- v_n^m : la vitesse du véhicule n à la date t_m.

L'ensemble des positions des N véhicules au cours des M dates est rassemblé dans la matrice L de taille (N, M), telle que :

- chaque ligne contient les positions successives du même véhicule au cours du temps ;
- chaque colonne indique la position de tous les véhicules à une date donnée.

$$\begin{array}{rcl}
 & \begin{array}{cccccc} t_0 & t_1 & \dots & t_m & \dots & t_{M-1} \\ \downarrow & \downarrow & & \downarrow & & \downarrow \end{array} & \\
L = & \begin{pmatrix} x_0^0 & x_0^1 & \dots & x_0^m & \dots & x_0^{M-1} \\ x_1^0 & x_1^1 & \dots & x_1^m & \dots & x_1^{M-1} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ x_n^0 & x_n^1 & \dots & x_n^m & \dots & x_n^{M-1} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ x_{N-1}^0 & x_{N-1}^1 & \dots & x_{N-1}^m & \dots & x_{N-1}^{M-1} \end{pmatrix} & \begin{array}{l} \longleftarrow \text{véhicule de tête } n = 0 \\ \\ \\ \longleftarrow \text{véhicule } n \\ \\ \longleftarrow \text{dernier véhicule } n = N-1 \end{array}
\end{array}$$

Le **code Python 2** permet de générer la matrice L en utilisant le modèle proposé. Les parties repérées par « **___** » seront à remplir.

```
1  import numpy as np
2  from enregistrements import positions          # importation des données
3
4  # Définition de la fonction f
5  def f(d, dmin=7.0, d0=20):                     # Voir Q24
6      if ___:
7          return ___
8      else:
9          return ___
10
11 # Initialisation de la matrice L
12 def initialisation(positions, N, d):
13     M = ___                                    # Voir Q25
14     L = ___                                    # Voir Q26
15     L[0, :] = positions                        # Voir Q27
16     for n in range(1, N):
17         L[n, 0] = ___                          # Voir Q28
18     return L
19
20 # Déplacement des véhicules
21 def deplacement(L, f, vmax, Dt):
22     N = len(L)        # Nombre de lignes de L
23     M = len(L[0])     # Nombre de colonnes de L
24     for m in range(___):                       # Voir Q30
25         for n in range(___, ___):              # Voir Q31
26             d = L[n - 1, m] - L[n, m]          # Voir Q32
27             L[n, m + 1] = ___                  # Voir Q33
28     return
29
30 L = initialisation(positions, 10, 30)
31 deplacement(L, f, 30, 1)
```

Code Python 2 - Effet d'un ralentissement

Les positions x_0 du véhicule de tête ont été enregistrées à intervalles de temps réguliers. Ces données sont stockées dans le tableau `positions` importé par le module `enregistrements`, comme le montre le code suivant.

```
>>> from enregistrements import positions
>>> positions
array([  0,  20,  40,  60,  80,  90,  95, 100, 105, 110, 130, 150, 170,
       190, 210, 230, 250, 270, 290, 310, 330])
```

Définition de la fonction f

Q24. En utilisant l'**équation (7)** définissant la fonction f, compléter le **code Python 2** à partir de la **ligne 5**. On utilisera les variables `dmin` et `d0` utilisables dans le corps de la fonction sans avoir besoin d'être définies.

Initialisation de la matrice L

La fonction `initialisation()` du **code Python 2** permet de définir et d'initialiser la matrice L. Elle prend comme paramètres le tableau des positions du véhicule de tête, le nombre total N de véhicules et la distance d les séparant à la date t_0.

Q25. Compléter la **ligne 13** indiquant le nombre de dates que contient la variable `positions`.

Q26. En utilisant une commande de l'annexe, compléter la **ligne 14** permettant d'initialiser une matrice L, de N lignes et M colonnes, remplie de zéros.

Q27. Expliquer ce qui est fait à la **ligne 15** du **code Python 2**.

Initialement, l'écart entre les véhicules est identique et égal à d. Le premier est à la position $x_0 = 0$, le deuxième est à la position $x_1 = -d$, le troisième à la position $x_2 = -2d$, etc.

Q28. Compléter la **ligne 17** permettant de remplir la 1re colonne de la matrice L avec les positions initiales des N véhicules.

Déplacement des véhicules

Le déplacement des véhicules est régi par l'équation :

$$\frac{\mathrm{d}x_n^m}{\mathrm{d}t} = v_{\max} \times f(x_{n-1}^m - x_n^m). \qquad \textbf{(8)}$$

En notant Dt le pas de temps numérique, on peut faire l'approximation :

$$\frac{\mathrm{d}x_n^m}{\mathrm{d}t} = \frac{x_n^{m+1} - x_n^m}{Dt}. \qquad \textbf{(9)}$$

Q29. En utilisant les équations **(8)** et **(9)**, établir l'expression de x_n^{m+1} en fonction de x_n^m, x_{n-1}^m, $v_{\max}$, f et de Dt.

La fonction `deplacement()` du **code Python 2** permet d'implémenter la relation de récurrence définie à la question **Q29** et de compléter ainsi la matrice L pour rendre compte du déplacement des véhicules au cours du temps.

On rappelle que la commande `range(a, b)` donne accès aux entiers compris dans l'intervalle $[\![a, b-1]\!]$.

Q30. Compléter la **ligne 24** à l'aide de la variable M.

Q31. Compléter la **ligne 25** à l'aide de la variable N.

Q32. Expliquer ce que calcule la **ligne 26**.

Q33. Compléter la **ligne 27** en utilisant tout ce qui est nécessaire pour implémenter la relation de récurrence obtenue à la question **Q29**.

La représentation graphique de la matrice L est donnée sur la **figure 7**. On peut y voir l'évolution de la position (sur l'axe y) de $N = 10$ véhicules au cours du temps (sur l'axe des x). Les positions du véhicule de tête sont reliées par un trait continu.

Q34. Donner l'intervalle des dates correspondant au ralentissement du véhicule de tête.
Déterminer, en le justifiant, le sens de propagation du ralentissement (sens des x croissants ou des x décroissants).
Estimer la vitesse de propagation du ralentissement.

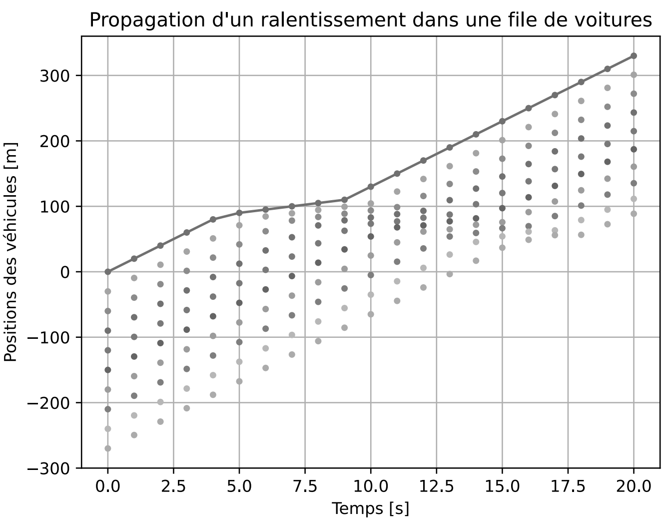

Figure 7 - Propagation d'un ralentissement dans une file de voiture

Partie III - Étude de la batterie d'une voiture électrique

La part des véhicules électriques, notamment les voitures, est en augmentation importante. Nous allons nous intéresser à la batterie d'une « Tesla Model 3 », qui est constituée de plusieurs milliers de cellules électrochimiques de type lithium–ion.

III.1 - Étude d'une cellule électrochimique d'une batterie lithium–ion

Le numéro atomique du lithium est $Z = 3$.

Q35. Indiquer la position de l'élément lithium dans la classification périodique des éléments chimiques. Préciser la période et la famille.

Q36. L'ion lithium le plus stable est Li^+. Justifier.

Q37. Préciser l'intérêt électrochimique majeur de l'élément lithium.

Description d'une cellule

De façon simplifiée, nous supposons qu'une cellule électrochimique est constituée (**figure 8**) :

- d'une *électrode* de graphite dans laquelle un élément lithium peut s'insérer pour former du LiC_6 ;
- d'une *électrode* d'oxyde de nickel NiO_2 dans laquelle un élément lithium peut s'insérer pour former du $LiNiO_2$;
- d'un *séparateur* entre l'anode et la cathode, qui est une membrane plastique fine microperforée perméable notamment aux ions Li^+.

L'ensemble baigne dans un gel électrolytique assurant le transport des ions lithium d'une électrode à l'autre.

Selon que l'accumulateur fonctionne en charge ou en décharge, les électrodes alternent leur rôle d'anode et de cathode.

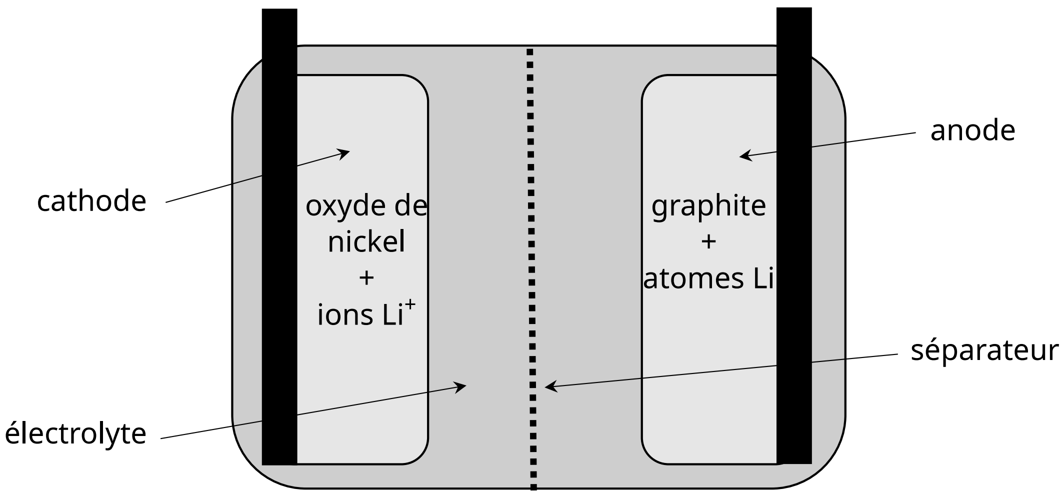

Figure 8 - Schéma d'une cellule lithium–ion

Charge de la batterie

À l'électrode de graphite, on observe la réduction des ions lithium.

Q38. Écrire la demi-équation électronique de réduction des ions lithium en atome lithium.

Le lithium formé s'insère dans l'électrode de graphite, ce que l'on traduit par l'équation de réaction : $Li + C_6 \;=\; LiC_6$.

Q39. En déduire la demi-équation d'oxydoréduction mettant en jeu Li^+ et LiC_6.

À l'électrode d'oxyde de nickel lithié $LiNiO_2$, on observe le mécanisme inverse : il y a désinsertion d'un ion lithium Li^+ et formation de l'oxyde de nickel NiO_2.

Q40. Écrire la demi-équation d'oxydoréduction ayant lieu à l'électrode de $LiNiO_2$ et produisant un ion lithium et de l'oxyde de nickel.

Q41. En déduire l'équation globale de fonctionnement de l'accumulateur.

La *capacité électrique massique* d'une batterie lithium–ion est la charge électrique maximale pouvant être délivrée par gramme de $LiNiO_2$.

Q42. Déterminer la capacité électrique massique q_m de l'accumulateur en $A\cdot h\cdot g^{-1}$.
Masse molaire $M(LiNiO_2) = 98\,g\cdot mol^{-1}$. Constante de Faraday $F = 96\cdot 10^3\,C\cdot mol^{-1}$.
Conversion : $1\,A\cdot h = 3\,600\,C$.

Q43. La charge maximale mesurée vaut $Q_{max} = 3{,}9\,A\cdot h$.
En utilisant la valeur de q_m de la question **Q42**, déterminer la masse de $LiNiO_2$ contenue dans la cellule.

La batterie de la « Tesla Model 3 » est formée de 96 modules placés en série, chaque module comportant 46 cellules électrochimiques placées en parallèle. La force électromotrice d'une cellule vaut 3,7 V.

Q44. Déterminer la tension aux bornes de la batterie.

Structure cristalline de l'électrode de $LiNiO_2$

La cristallisation du $LiNiO_2$ respecte les conditions suivantes :

- les ions oxygène O^{2-} forment un réseau cubique à faces centrées ;
- les ions nickel Ni^+ occupent une partie des sites octaédriques ;
- les ions lithium Li^+ occupent une partie des sites tétraédriques.

Q45. Indiquer, à l'aide d'un vocabulaire spécifique, la position des ions oxygène dans une maille, ainsi que celle des sites tétraédriques et des sites octaédriques.

Q46. Déterminer le nombre d'ions oxygène par maille.

Les ions nickel et lithium sont en proportions égales.

Q47. Déterminer le nombre d'ions nickel par maille et le nombre d'ions lithium par maille.
En déduire le pourcentage d'occupation des sites octaédriques par les ions nickel et le pourcentage d'occupation des sites tétraédriques par les ions lithium.

Le rayon ionique de l'oxygène vaut $R = 140$ pm et celui de l'ion nickel vaut $r = 64{,}0$ pm.

Q48. Dans l'hypothèse où les cations sont tangents aux anions, calculer le paramètre de la maille.

Q49. Établir l'expression du rayon du plus gros cation que l'on puisse insérer dans un site tétraédrique sans déformer la maille.
En déduire si le cristal est déformé par la présence de l'ion lithium de rayon 60,0 pm.

III.2 - Charge maximale d'une cellule

On se propose de mesurer la charge maximale d'une cellule lithium–ion lors d'une charge complète. La charge comporte deux phases, comme le montre la **figure 9** :

- la première phase, rapide, s'effectue à intensité I constante et égale à 2,8 A, jusqu'à ce que la tension aux bornes de la cellule atteigne 3,7 V ;
- la seconde phase, plus lente, s'effectue à tension U constante égale à 3,7 V.

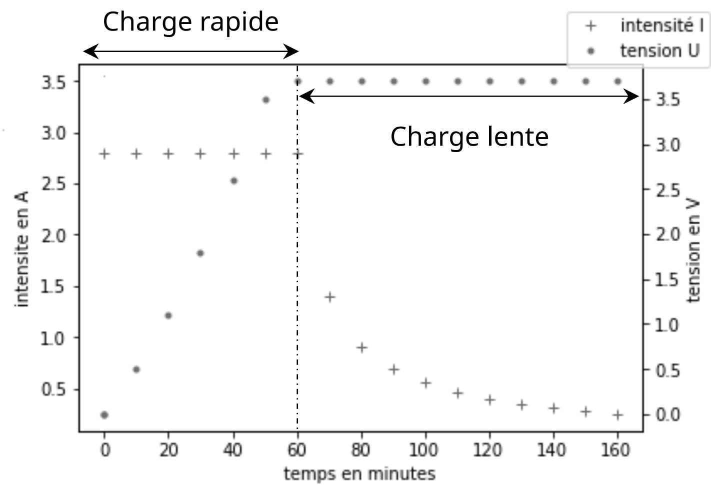

Figure 9 - Évolution de l'intensité et de la tension de la cellule au cours du temps

Dispositif de charge

Pour réaliser la 1[re] phase de charge de la cellule, à intensité constante, on la place dans un dispositif partiellement représenté sur la **figure 10**, permettant de contrôler l'intensité I et la tension U aux bornes de la cellule électrochimique.

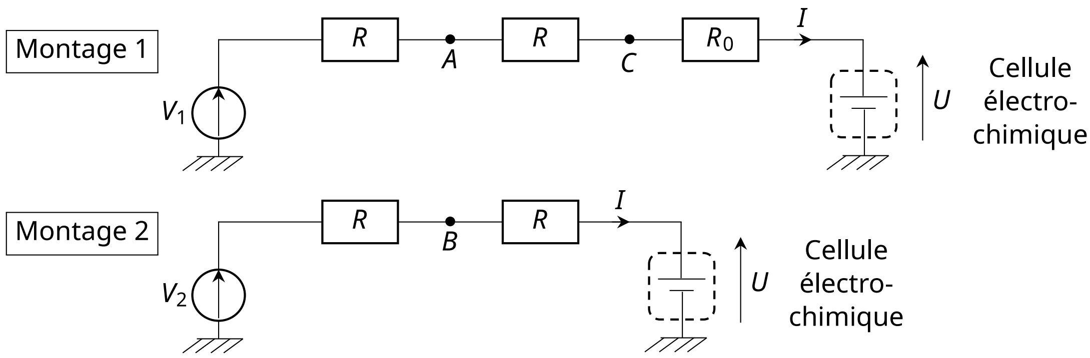

Figure 10 - Dispositif de charge

Q50. À partir du montage 1, exprimer le potentiel V_A en A en fonction de la tension V_1 et du potentiel V_C au point C.

Q51. À partir du montage 1, exprimer le potentiel V_C en fonction de U, I et de R_0.

Q52. À partir du montage 2, exprimer le potentiel V_B en B en fonction de U et de V_2.

Une partie non représentée du montage assure l'égalité des tensions en A et B, sans circulation de courant entre A et B. On dispose donc de la relation $V_A = V_B$.

Q53. En déduire l'expression de l'intensité I en fonction de V_1, V_2 et de R_0.
Préciser la condition portant sur V_1 et V_2 permettant de réaliser une charge à intensité I constante.

Pour réaliser la seconde phase de charge de la cellule, à tension constante, un microcontrôleur non représenté fait varier les tensions V_1 et V_2 de manière à maintenir la tension U constante. Cette étape n'est pas étudiée.

Exploitation expérimentale

Q54. Donner la relation entre l'intensité $I(t)$ qui traverse la cellule et la charge $Q(t)$ qu'elle contient.

L'intensité circulant lors de la charge a été enregistrée au cours du temps à différentes dates pour un ensemble discret de dates. On note $Q_k = Q(t_k)$ la valeur de la charge à la date t_k et $I_k = I(t_k)$ celle de l'intensité à la date t_k.

Q55. Donner la relation entre Q_{k+1}, Q_k et $\displaystyle\int_{t_k}^{t_{k+1}} I(t)\mathrm{d}t$.

Pour calculer l'intégrale de la fonction f continue, on utilise la relation approchée suivante illustrée sur la **figure 11** :

$$\int_a^b f(x)\mathrm{d}x \approx \frac{1}{2}(b-a)\big(f(a)+f(b)\big).$$

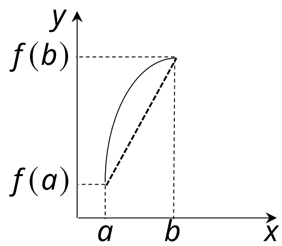

Figure 11 - Méthode des trapèzes

Q56. Donner l'expression approchée de $\displaystyle\int_{t_k}^{t_{k+1}} I(t)\mathrm{d}t$ en fonction de I_k, I_{k+1}, t_k et de t_{k+1}.
En déduire une estimation de Q_{k+1} en fonction de Q_k, I_k, I_{k+1}, t_k et de t_{k+1}.

On souhaite écrire une fonction Python qui calcule la charge maximale de la cellule. On dispose pour cela des deux listes suivantes correspondant aux mesures faites sur la cellule :

- `intensite` : qui contient l'intensité de charge en ampère,
- `date` : qui contient les dates des mesures en heure.

Q57. Écrire une fonction Python, de signature :

```
chargeMax(intensite: list, date: list) -> float
```

qui admet comme paramètres les listes `intensite` et `date` et renvoie la charge finale en A·h atteinte par la cellule. Il s'agit de la charge maximale.
(le résultat de cette fonction a été utilisé à la question **Q42**)

ANNEXE

Bibliothèque `numpy` de Python

Import de la bibliothèque **`numpy`** :

```
>>> import numpy as np
```

Création d'une matrice `M` de 2 lignes et 3 colonnes, remplie de zéros :

```
>>> M = np.zeros((2, 3))  # génère une matrice de 0
>>> print(M)
[[0. 0. 0.]
 [0. 0. 0.]]
```

Modification de cellules, lignes ou colonnes de la matrice `M` (les indices démarrent à 0) :

```
>>> M[0, 0] = 5
>>> M[1, 2] = 1
>>> print(M)
[[5. 0. 0.]
 [0. 0. 1.]]
>>> print(M[1, :])  # affiche la ligne 1
[0. 0. 1.]
>>> M[1, :] = 3     # modifie toutes les colonnes de la ligne 1
>>> print(M)
[[5. 0. 0.]
 [3. 3. 3.]]
>>> print(M[:, 0])  # affiche la colonne 0
[5. 3.]
>>> M[:, 2] = -7    # modifie toutes les lignes de la colonne 2
>>> print(M)
[[5. 0. -7.]
 [3. 3. -7.]]
```

Fonction mathématique : exponentielle

```
>>> np.exp(1)    # la fonction exponentielle
2.718281828459045
```

Commandes Python : **`append()`**, **`sum()`** et **`len()`**

```
>>> L = [1, -5]
>>> L.append(7)  # ajoute un élément à la fin de la liste L
>>> L
[1, -5, 7]
>>> sum(L)       # calcule la somme des éléments de la liste L
3
>>> L.append(-17)
>>> L
[1, -5, 7, -17]
>>> len(L)       # donne le nombre d'éléments de la liste L
4
```

Bibliothèque **matplotlib** de Python

Import de la sous-bibliothèque **pyplot** de la bibliothèque **matplotlib** :

```
>>> import matplotlib.pyplot as plt
```

Aide de la commande **plt.bar** pour le tracé d'un histogramme :

```
>>> help(plt.bar)
Help on function bar in module matplotlib.pyplot:

bar(x, height, width=0.8, bottom=None, *, align='center', data=None, **kwargs)
    Make a bar plot.

    The bars are positioned at *x* with the given *align*\ment. Their
    dimensions are given by *height* and *width*. The vertical baseline
    is *bottom* (default 0).

    Many parameters can take either a single value applying to all bars
    or a sequence of values, one for each bar.

    Parameters
    ----------
    x : float or array-like
        The x coordinates of the bars. See also *align* for the
        alignment of the bars to the coordinates.

    height : float or array-like
        The height(s) of the bars.

    width : float or array-like, default: 0.8
        The width(s) of the bars.

    bottom : float or array-like, default: 0
        The y coordinate(s) of the bottom side(s) of the bars.

    align : {'center', 'edge'}, default: 'center'
        Alignment of the bars to the *x* coordinates:

        - 'center': Center the base on the *x* positions.
        - 'edge': Align the left edges of the bars with the *x* positions.

        To align the bars on the right edge pass a negative *width* and
        "align='edge'".
```

FIN

IMPRIMERIE NATIONALE – 24 1118 – D'après documents fournis