Lycée BLAISE PASCAL

Concours blanc – Épreuve d'Informatique et Modélisation

Durée : 4 heures

Si, au cours de l'épreuve, un candidat repère ce qui lui semble être une erreur d'énoncé, il le signale sur sa copie et poursuit sa composition en précisant les raisons des initiatives qu'il est amené à prendre.

L'usage de la calculatrice est interdit.

Avertissements aux candidats :

La présentation, la lisibilité, l'orthographe, la qualité de la rédaction, la clarté et la précision des raisonnements entreront pour une part importante dans l'appréciation des copies. **En particulier, les résultats non justifiés ne seront pas pris en compte.** Les candidats doivent encadrer les résultats de leurs calculs.

Les fonctions et programmes doivent être écrits en langage Python ou SQL. On prendra garde à la **lisibilité du code**, et notamment aux indentations. Le code devra être **documenté**. Toute fonction définie dans l'énoncé pourra être utilisée même si elle n'a pas été codée.

Les candidats veilleront à traiter la partie I d'une part et les parties II et III d'autre part sur des **copies indépendantes**.

Des **annexes** sur le fonctionnement de certaines fonctions Python sont fournies en fin de sujet.

Les candidats doivent vérifier que le sujet comporte bien 12 pages, numérotées de 1/12 à 12/12.

Fusible

Ce problème aborde le fonctionnement des fusibles, développés à partir du XVIII^{e} siècle pour sécuriser les expériences de l'époque sur l'électrisation, en particulier avec des condensateurs rudimentaires qui avaient donné lieu à plusieurs accidents marquants. La structure de base a peu évolué depuis, et un fusible présente toujours les mêmes éléments essentiels représentés figure 1 :

- ▷ deux pièces de connexion appelées plots permettant de relier le fusible au reste du circuit électrique ;
- ▷ un fil conducteur fait d'un métal à faible point de fusion ;
- ▷ une gaine qui assure un rôle de protection électrique et peut contenir un isolant thermique.

Le principe de fonctionnement est très simple. En fonctionnement normal, le fusible assure le passage du courant. Si un défaut électrique apparaît et créée un courant anormalement élevé, le fil métallique s'échauffe et fond, ce qui permet une ouverture automatique du circuit électrique.

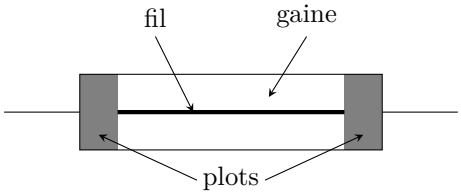

Figure 1 – Schéma de principe d'un fusible.

I - Évolution de la température au sein d'un fusible

I.A - Effet Joule dans le fusible

On considère un fusible dont le fil métallique est de forme cylindrique d'axe (Oz), de section constante $S = 1 \cdot 10^{-7}\,\mathrm{m}^2$, de longueur $L = 3\,\mathrm{cm}$ et de conductivité électrique σ. Une différence de potentiel constante $U = V(z=L) - V(z=0)$ est appliquée entre ses extrémités. On suppose les équipotentielles planes et perpendiculaires à l'axe (Oz).

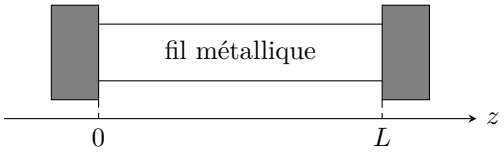

Figure 2 – Notations pour l'étude du fusible.

1 - Rappeler l'équation de Maxwell-Gauss et démontrer l'équation de Poisson.

2 - Admettant que le métal est localement neutre, en déduire que le champ électrostatique dans le fusible s'écrit

$$\vec{E} = -\frac{U}{L}\vec{e}_z\,.$$

Le métal est modélisé par un réseau cristallin d'ions positifs fixes dans lequel des électrons de conduction, de charge $-e$ et de masse $m_\mathrm{e} = 1 \cdot 10^{-30}\,\mathrm{kg}$ se déplacent librement. La densité volumique d'électrons de conduction dans le métal est notée n. Le mouvement des électrons est étudié dans la théorie de Drude, où les interactions entre les électrons de conduction et le réseau cristallin sont modélisées par une unique force

$$\vec{f} = -\frac{m_\mathrm{e}}{\tau}\vec{v}\,,$$

où τ est un paramètre phénoménologique nommé temps de relaxation.

3 - Montrer que le poids d'un électron est négligeable dans les conditions d'utilisation du fusible ($U \sim 3\,\mathrm{V}$).

4 - Déterminer la vitesse $\vec{v}_\infty$ d'un électron en régime permanent.

5 - On rappelle que la densité volumique de courant est reliée à la vitesse des électrons de conduction par $\vec{j} = -ne\vec{v}_\infty$. Définir la conductivité électrique σ du matériau, et l'exprimer en fonction des paramètres du modèle.

La force de frottement subie par les électrons traduit une dissipation d'énergie dans le fil : c'est l'effet Joule. On cherche à estimer la puissance $\mathrm{d}\mathcal{P}_\mathrm{J}$ dissipée par effet Joule dans un tronçon élémentaire de fil de longueur $\mathrm{d}z$, qui est simplement la somme des puissances de frottement subies par les électrons s'y trouvant.

6 - Exprimer la puissance de la force de frottement subie par un électron, et en déduire une expression de $\mathrm{d}\mathcal{P}_\mathrm{J}$ en fonction notamment de v_∞.

7 - Montrer enfin que

$$\mathrm{d}\mathcal{P}_\mathrm{J} = \frac{I^2}{\sigma S}\mathrm{d}z\,.$$

I.B - Profil de température dans un fusible de section uniforme

On s'intéresse à la distribution de température $T(z)$ dans le fusible, dont on note λ la conductivité thermique, $T^\star$ la température de fusion et c la capacité thermique massique. La gaine est supposée parfaitement isolante thermiquement. Les plots sont assimilés à des conducteurs électriques et thermiques parfaits, de température constamment égale à la température extérieure T_0. On se place en régime permanent.

8 - On se place dans l'hypothèse d'équilibre thermodynamique local. Expliquer en quoi cette approximation consiste et pourquoi elle est pertinente ici.

9 - Rappeler la loi de Fourier, ses hypothèses, et son interprétation physique. Donner les unités de chaque terme.

10 - Montrer que l'équation différentielle vérifiée par la température s'écrit

$$\frac{\mathrm{d}^2T}{\mathrm{d}z^2} + \frac{I^2}{\lambda\sigma S^2} = 0\,.$$

11 - En déduire le profil de température $T(z)$, faisant intervenir deux constantes A et B.

12 - Déterminer les constantes A et B en justifiant. Conclure sur l'expression de $T(z)$.

13 - Représenter qualitativement l'allure de $T(z)$. Le tracé sera justifié par des arguments qualitatifs précis.

14 - Déterminer le courant de coupure pour lequel le fusible ouvre le circuit.

La question 15 ci-dessous n'est pas guidée et demande de l'initiative de la part du candidat. Toutes les pistes explorées par le candidat doivent être rapportées sur la copie avec les hypothèses associées, même si elles n'aboutissent pas : dès lors qu'elles sont pertinentes, elles seront récompensées. Cette question est fortement valorisée par le barème ; toutefois il est conseillé au candidat de ne pas y consacrer plus de quinze minutes.

15 - Estimer un ordre de grandeur du temps de réponse du fusible, c'est-à-dire le temps nécessaire pour qu'il ouvre le circuit à partir du moment où apparaît un défaut électrique. Pour simplifier, on pourra considérer la température au sein du fusible uniforme par morceaux. Un raisonnement littéral argumenté est attendu, ainsi qu'une valeur numérique pour laquelle le candidat proposera des ordres de grandeur pour les différents paramètres qu'il aura été amené à introduire. On prendra un courant de 10 A.

I.C - Équation de la chaleur dans un fusible de section variable

En réalité, la géométrie précédente ne permet pas de satisfaire aux exigences de sécurité : le courant de coupure est très élevé, le temps de réponse trop important, et l'ouverture du circuit entraîne des phénomènes de surtension pouvant être néfastes pour les équipements à protéger. Une amélioration consiste à introduire des rétrécissements de section, appelés crans, régulièrement espacés le long du fusible, comme représenté figure 3. Ce faisant, tous les crans s'ouvrent simultanément, ce qui permet d'atténuer notamment les surtensions. En pratique, il ne s'agit plus d'un fil cylindrique, mais d'une lame d'épaisseur uniforme et de largeur variable.

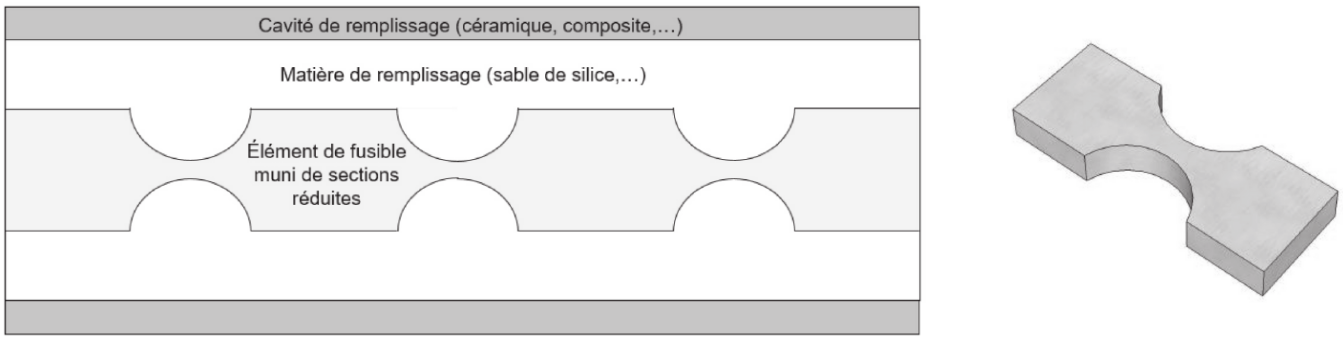

Figure 3 – Schéma d'un fusible à crans.

On s'intéresse à l'effet d'un seul rétrécissement, situé entre les abscisses $z = 0$ et $z = L$. La température est supposée homogène dans toute section perpendiculaire à l'axe du fusible : on a donc $T = T(z)$ en régime permanent, et le vecteur densité de flux thermique s'écrit

$$\vec{j}_q = j_q(z)\,\vec{e}_z\,.$$

16 - Par un bilan thermique sur un tronçon élémentaire de fusible, montrer que

$$\frac{\mathrm{d}}{\mathrm{d}z}\Big(S(z)\,j_q(z)\Big) = \frac{I^2}{\sigma S(z)}\,.$$

17 - En déduire que la température vérifie l'équation différentielle

$$\frac{\mathrm{d}^2T}{\mathrm{d}z^2} + \frac{1}{S(z)}\frac{\mathrm{d}S}{\mathrm{d}z}\frac{\mathrm{d}T}{\mathrm{d}z} = -\frac{I^2}{\lambda\sigma\,S(z)^2}$$

II - Étude numérique d'un fusible de section variable

L'équation différentielle obtenue dans la partie précédente est non linéaire et ne peut être résolue à la main : on se tourne donc vers une approche numérique. Pour ce faire, on l'écrit sous la forme

$$\frac{\mathrm{d}^2T}{\mathrm{d}z^2} + g(z)\,\frac{\mathrm{d}T}{\mathrm{d}z} = -\frac{k}{S(z)^2} \qquad \text{avec} \qquad \begin{cases} g(z) = \dfrac{1}{S(z)}\dfrac{\mathrm{d}S}{\mathrm{d}z} \\ k = \dfrac{I^2}{\lambda\sigma} \end{cases}$$

Le rétrécissement de la section du fil est modélisé par la fonction

$$S(z) = S_0\left(1 - a\sin\frac{\pi z}{L}\right) \qquad \text{avec} \qquad \begin{cases} S_0 = 0{,}105\,\mathrm{mm}^2 \\ a = 0{,}9 \\ L = 1{,}66\,\mathrm{mm} \end{cases}$$

On suppose la température imposée aux deux extrémités :

$$T(z{=}0) = T(z{=}L) = T_0 = 293\,\mathrm{K}\,.$$

Les modules importés et variables globales prédéfinies sont les suivants :

```
import numpy as np
import matplotlib.pyplot as plt

from scipy.integrate import odeint

S0 = 0.105e-6    # m2
a = 0.9
L = 1.66e-3      # m
k = 1e-6         # K.m2
T0 = 293         # K
```

II.A - Résolution numérique

On cherche à déterminer une liste de $n = 1000$ éléments donnant le profil de température pour des valeurs de z équiréparties entre 0 et L inclus. On pose $T_i = T(z_i)$ pour $0 \leq i \leq n-1$.

18 - Donner l'expression de la position z_i en fonction de i, n et L.

19 - Justifier qualitativement l'expression choisie pour $S(z)$, et calculer la fonction g associée.

20 - Écrire une fonction `S` prenant en paramètre l'abscisse `z` et renvoyant la valeur de la section à la position z. On suppose disposer d'une fonction `g` jouant un rôle analogue, qu'il n'est pas demandé de coder.

L'équation différentielle que l'on cherche à résoudre est en fait une équation différentielle du premier ordre portant sur la dérivée première de la température. On pose

$$v = \frac{\mathrm{d}T}{\mathrm{d}z} \qquad \text{et} \qquad v_i = \frac{\mathrm{d}T}{\mathrm{d}z}(z{=}z_i)\,.$$

21 - Expliquer le principe de la méthode d'Euler permettant l'intégration numérique de l'équation différentielle du premier ordre étudiée.

22 - Écrire le code mettant en œuvre la méthode décrite à la question précédente, et produisant la liste `v` dont l'élément `v[i]` donne la dérivée de la température v_i. On prendra comme condition limite $v_0 = 0$.

23 - Écrire le code produisant la liste `T_Euler` des températures à partir de la liste `v`.

Pour résoudre l'équation avec une meilleure précision et un moindre temps de calcul, il est préférable d'utiliser la fonction `odeint` d'intégration numérique d'équation différentielle, présente dans le module `scipy.integrate`. Le fonctionnement de cette fonction est détaillé en annexe.

24 - Écrire l'équation différentielle à résoudre sous la forme

$$\frac{\mathrm{d}\,\vec{u}}{\mathrm{d}z} = \vec{f}(\vec{u}(z), z)$$

en identifiant le vecteur $\vec{u}$ et la fonction $\vec{f}$ pour le problème étudié ici.

25 - En utilisant les notations de l'annexe, écrire dans le cadre du problème étudié la fonction `F` qui renvoie la dérivée seconde de la température pour la position `z`. Le candidat sera vigilant sur les arguments attendus.

26 - Créer la liste `T_odeint` contenant $n = 1000$ valeurs de la température entre $z = 0$ et $z = L$. La dérivée première sera prise nulle en $z = 0$.

27 - Écrire le code permettant de représenter la température ainsi calculée en fonction de l'abscisse z. On pensera à légender les axes de la figure.

28 - Le tracé obtenu est reproduit figure 4. Justifier que ce résultat ne peut pas correspondre au problème physique étudié.

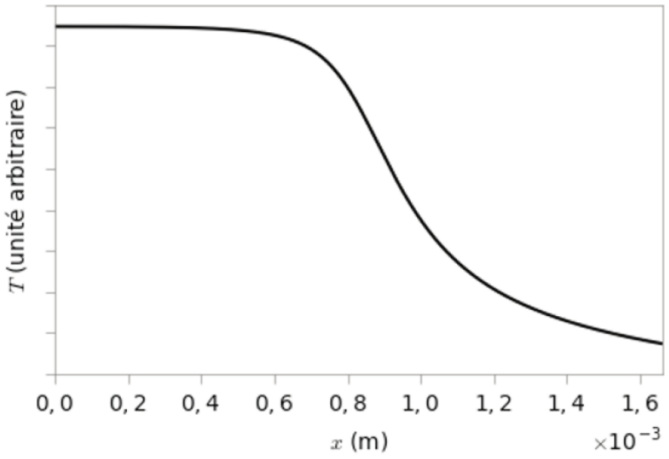

Figure 4 – **Profil de température dans un fusible de section variable.** Profil obtenu avec une condition limite nulle sur la dérivée.

II.B - Recherche des conditions aux limites

Le problème soulevé à la partie précédente vient d'un mauvais choix de condition limite sur la dérivée de la température. Pour l'identifier, on procède par dichotomie : la bonne valeur est celle qui permet d'avoir la même température T_0 en $z = 0$ et $z = L$.

29 - Expliquer le principe de la méthode dichotomique, permettant de déterminer un zéro d'une fonction sur un intervalle $[a, b]$ connu. On pourra illustrer les explications d'un schéma.

30 - Écrire une fonction `TL(T0,v0)` qui prend comme argument les deux conditions aux limites sur la température T_0 et sa dérivée v_0 en $z = 0$ et qui renvoie la valeur de la température en $z = L$. La liste `z` des positions est supposée accessible en tant que variable globale.

31 - Écrire une fonction `dicho_derivee(T0, epsilon, v_min, v_max)` renvoyant l'estimation de la dérivée de la température v_0 en $z = 0$ permettant d'avoir une température T_0 en $z = L$. Cette fonction prend comme paramètres :

- ▷ la température `T0` en $z = 0$;
- ▷ l'écart `epsilon` toléré sur l'égalité des températures en $z = 0$ et $z = L$;
- ▷ les valeurs extrêmes `v_min` et `v_max` entre lesquelles on cherchera la valeur de la dérivée.

On notera également que la valeur de $T(z{=}L)$ augmente lorsque $v(z{=}0)$ augmente.

II.C - Exploitation du résultat

32 - En utilisant la condition limite déterminée par dichotomie, on obtient le profil de température de la figure 5. Commenter la courbe obtenue.

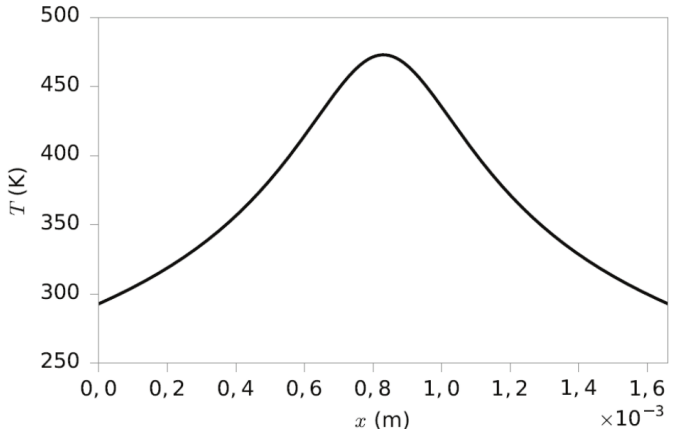

Figure 5 – **Profil de température dans un fusible de section variable.** Profil obtenu avec une condition limite sur la dérivée obtenue par dichotomie.

33 - Écrire une fonction `recherche_max(L)` renvoyant le plus grand élément d'une liste `L`. On n'utilisera pas la fonction **max** de Python.

34 - On trace alors sur la figure 6 la température maximale au sein du fusible en fonction du courant qui le traverse, sans prendre en compte le phénomène de fusion. Commenter la courbe obtenue, et en particulier la nature des asymptotes.

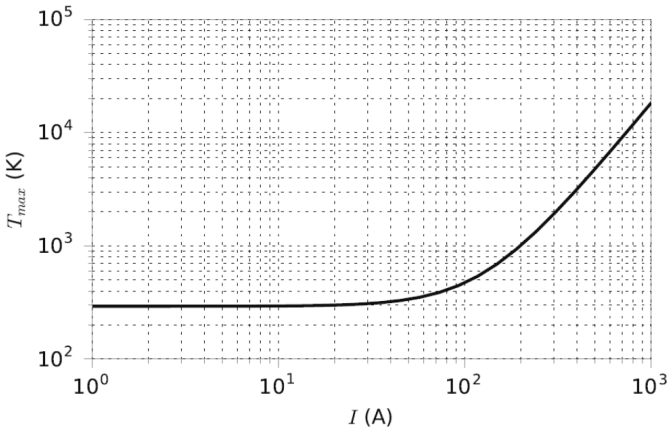

Figure 6 – **Température maximale au sein du fusible.**

III - Une base de données

Dans toute cette partie, on considère une base de données dont la structure est la suivante :

▷ la table **fusibles** contient les attributs suivants :

Attribut	Type de donnée
id	un nombre, identifiant unique
type	texte, contient l'un de "uniforme" ou "variable"
coupure	nombre flottant, le courant de coupure en ampères
fusion	nombre entier, la température de fusion en kelvins

▷ la table **clients** contient les attributs suivants :

Attribut	Type de donnée
id	un nombre, identifiant unique
nom	texte
...	

▷ la table **commandes** contient les attributs suivants :

Attribut	Type de donnée
id	un nombre, identifiant unique
id_client	identifiant du client
id_fusible	identifiant du fusible
nombre	un entier, le nombre de fusible commandé
date	du texte, au format "AAAA-MM-JJ", la date de la commande

III.A - Quelques requêtes

35 - Écrire une requête permettant de récupérer les types de fusible, courant de coupure et température de fusion pour les fusibles dont le courant de coupure est au moins 10A.

36 - Identifier précisément les attributs des tables permettant d'effectuer des jointures.

37 - Donner toutes les dates de commande du client dont le nom est "Nonyme" (On suppose que le nom identifie de manière unique ce client).

38 - Ce même client a effectué une commande le mercredi 1er mars 2023. Donner le nombre de fusibles commandés, leurs types ainsi que leurs courants de coupure.

III.B - Exploitation des résultats

On se place maintenant dans le cadre où le résultat d'une requête a été récupéré en python. Plus précisément, on considère que l'on dispose d'une liste `commandes` qui regroupe toutes les informations des commandes sur une période donnée.

Les éléments de la liste `commandes` sont des dictionnaires qui ont tous les mêmes clés. Voici un exemple d'un tel dictionnaire :

```
{"nom": "Nonyme",
"date": "2023-02-05",
"nombre": 5,
"type": "variable",
"coupure": 10,
"fusion": 1300}
```

Lorsque qu'un argument d'une des fonctions suivantes est nommé `commandes`, la liste passée sera forcément sous cette forme.

39 - Écrire une fonction `nombre_total` qui prend comme argument une liste `commandes` au format précédent et retourne le nombre total de fusibles commandés.

40 - Écrire une fonction `type_majoritaire` qui prend le même argument et retourne la chaîne de caractère correspondant au type de fusible le plus commandé sur la période décrite par la liste `commandes`. En cas d'égalité, la valeur de retour est à votre convenance.

41 - Expliquer par une ou deux phrases quelle fonction et quelle agrégation on peut utiliser en SQL pour obtenir le résultat de la fonction précédente.

42 - Écrire une fonction `nombre_par_jour` qui prend comme argument une liste `commandes` et retourne le dictionnaire dont les clés sont des dates et les valeurs le nombre de fusibles commandés à cette date.

43 - Un problème de production des fusibles de type "uniforme" et ayant un courant de coupure de 10A a été détecté. Expliquer par quelques phrases comment construire la liste des clients à contacter sachant que l'on dispose d'une liste `commandes`

44 - Mettre en œuvre votre algorithme dans une fonction `alerte` qui prend 3 arguments : une liste `commandes`, une chaîne de caractère `typ` contenant le type de fusible et un nombre `coupure` contenant la puissance de coupure problématique.

45 - En notant n la longueur de la liste `commandes`, quel est l'ordre de grandeur du nombre d'opérations effectués par la fonction précédente ? On compte comme opération toute comparaison entre chaînes ou nombres et toute modification mémoire (changer la valeur d'une variable, d'un élément de liste…). On demande ici la complexité dans le pire des cas.

III.C - Livraisons

On considère dans cette partie le problème consistant à livrer des fusibles à différents endroits, en essayant d'optimiser le trajet.

On considère que les points de livraisons sont représentés par des points d'un plan rapporté à un repère orthonormé. On représente alors en python un point de livraison par une liste de deux flottants qui sont les coordonnées de ce point.

46 - Écrire une fonction distance qui prend deux points du plan comme arguments (ce sont donc deux listes de longueur 2) et retourne la distance entre ces deux points.

47 - On considère $N \geq 3$ points $A_1, \dots, A_N$.

- ▷ On appelle chemin reliant ces points une suite de $N+1$ points qui commence et fini au même point et qui contient tous les autres points exactement une fois. Par exemple $A_1 \to A_2 \cdots \to A_N \to A_1$ est l'un des chemins envisageable.
- ▷ La longueur d'un tel chemin est la somme des longueurs des N segments qui le composent. Pour l'exemple précédent, la longueur est donc $A_1A_2 + A_2A_3 + \cdots + A_NA_1$.

Justifier rapidement que dans le cas $N = 3$, tous les chemins sont de même longueur.

48 - Montrer que dans le cas général, il existe $(N-1)!$ chemins commençant en A_1 et $\frac{(N-1)!}{2}$ longueurs à calculer si on veut faire une recherche exhaustive de la plus petite longueur possible.

49 - On suppose maintenant $N > 3$ et on considère le chemin $A_1 \to A_2 \to A_3 \to A_1$ qui relie les 3 premiers points. Proposer un algorithme qui permette de déterminer le plus court chemin obtenu en insérant le point A_4 à un endroit.

Ne traiter cette question qu'en tout dernier. Toute piste de réflexion sera valorisée.

50 - Déduire de la question précédente une fonction `chemin(L)` qui prend comme argument une liste `L` contenant au moins 3 points et retourne un chemin reliant les points de L. On représente un chemin par la liste des indices des éléments de `L` correspondant.

Pour continuer sur l'exemple de la question 47, la liste retournée serait $[0, 1, \dots, N-1, 0]$.

ANNEXE 1

Méthode de résolution numérique d'une équation différentielle d'ordre 2 à l'aide de la bibliothèque scipy

Position du problème :

On cherche à résoudre une équation différentielle d'ordre 2 du type $\frac{\mathrm{d}^2y}{\mathrm{d}x^2} = F\left(\frac{\mathrm{d}y}{\mathrm{d}x}, y(x), x\right)$.
On dispose des conditions « initiales » $y(x_0) = y_0$ et $\frac{\mathrm{d}y}{\mathrm{d}x}(x_0) = y_0{}'$.

Principe :

Une équation différentielle d'ordre 2 peut être ramenée à un système d'équations différentielles d'ordre 1.

Procédure :

- On se donne la fonction python **`F`** désignant le second membre de l'égalité $\frac{\mathrm{d}^2y}{\mathrm{d}x^2} = F\left(\frac{\mathrm{d}y}{\mathrm{d}x}, y(x), x\right)$ ainsi que les conditions initiales enregistrées dans les variables **`y0`** et **`y0prime`**.
- On ramène l'équation différentielle d'ordre 2 à un système d'équations d'ordre 1.

  On pose : $u_0(x) = y(x)$ *et* $u_1(x) = \frac{\mathrm{d}y}{\mathrm{d}x}(x)$.

  Alors $\frac{\mathrm{d}^2y}{\mathrm{d}x^2} = F\left(\frac{\mathrm{d}y}{\mathrm{d}x}, y(x), x\right)$ *devient* $\begin{cases} \frac{\mathrm{d}u_0}{\mathrm{d}x} = u_1(x) \\ \frac{\mathrm{d}u_1}{\mathrm{d}x} = F(u_1(x), u_0(x), x) \end{cases}$
- Vectoriellement, cela se traduit par :

  On pose $\overrightarrow{u(x)} = \begin{cases} u_0(x) \\ u_1(x) \end{cases}$.

  On a donc $\frac{\mathrm{d}\vec{u}}{\mathrm{d}x} = \vec{f}(\overrightarrow{u(x)}, x)$ *avec* $\vec{f}\left(\overrightarrow{u(x)}, x\right) = \begin{cases} u_1(x) \\ F(u_1(x), u_0(x), x) \end{cases}$ *et* $\overrightarrow{u(x=0)} = \begin{cases} y_0 \\ y_0{}' \end{cases}$
- On résout ce système avec la fonction **`odeint`** de la bibliothèque **`scipy.integrate`** grâce au squelette de programme suivant :

Programme :

```
from scipy.integrate import odeint
import numpy as np

def fonction_f_vecteur(u,x):
      return [u[1],F(u[1],u[0],x)]

# n est le nombre de positions pour lesquels on cherche y

x = np.linspace(x_min,x_max,n)
solution = odeint(fonction_f_vecteur,[y0,y0prime],x)
```


Tournez la page S.V.P.

Solution :

La variable renvoyée par **`odeint`** enregistrée dans **`solution`** est une matrice de type ndarray comprenant les valeurs de la fonction et de sa dérivée :

$$\texttt{solution} = \begin{bmatrix} [y_0 & y'_0] \\ \vdots & \vdots \\ [y_{n-1} & y'_{n-1}] \end{bmatrix}$$

Ainsi, on accède à la *i*-ième valeur de la fonction solution de l'équation différentielle $y(x_i)$ par **`solution[i,0]`** et à la *i*-ième dérivée par **`solution[i,1]`**.

ANNEXE 2

Fonctions mathématiques générales définies dans numpy

Copie d'écran du site : https://docs.scipy.org/doc/numpy/reference/routines.math.html

Trigonometric functions

sin(x[, out])	Trigonometric sine, element-wise.
cos(x[, out])	Cosine element-wise.
tan(x[, out])	Compute tangent element-wise.

Exponents and logarithms

exp(x[, out])	Calculate the exponential of all elements in the input array.
log(x[, out])	Natural logarithm, element-wise.

ANNEXE 3

Exemple de script et de tracé associé avec matplotlib

Script :

```
import numpy as np
import matplotlib.pyplot as plt

x = np.linspace(0,2*np.pi,30)
y = np.cos(x)
plt.plot(x,y)
plt.xlim((0,2*np.pi))
plt.xlabel("abscisses")
plt.ylabel("ordonnees")

plt.show()
```

Tracé :